

COEN/ELEC 390

Final Submission

iSeat

iSeat, you seat, we all find a seat!

Prepared by:

Adnan Saab: 40075504

Farah Salhany: 40074803

Marwan Al-Ghaziri: 40126554

Samson Kaller: 40136815

Shahin Khalkhali: 40057384

Shayan Khalkhali: 40059491

Presented to:

Dr. William E. Lynch

Concordia University

INTRODUCTION	4
I. MISSION STATEMENT	5
II. FINAL PRODUCT BACKLOG	7
III. DESIGN DOCUMENT.....	10
4.1 ANDROID APPLICATION WIREFRAMES	10
4.2 SYSTEM ARCHITECTURE	12
4.3 HARDWARE ARCHITECTURE	13
4.3.1 <i>Hardware</i>	13
4.3.2 <i>Prototype</i>	15
IV. SOFTWARE ARCHITECTURE	20
4.4.1 ARDUINO SOFTWARE	20
V. USER MANUAL.....	23
VI. DEFINITION OF DONE.....	28
SPRINT 1	28
SPRINT 2	35
SPRINT 3	42
VII. TEST DOCUMENT.....	63
SPRINT 1	63
7.1 TEST PLAN 1: PROFILE ACTIVITY	63
7.1.1 <i>Summary</i>	63
7.1.2 <i>Test Cases</i>	63
7.2 TEST PLAN 2: ADD SENSOR DIALOG FRAGMENT	65
7.2.1 <i>Summary</i>	65
7.2.2 <i>Test Cases</i>	65
7.3 TEST PLAN 3: PROFILE ACTIVITY	68
7.3.1 <i>Summary</i>	68
7.3.2 <i>Test Cases</i>	68
7.4 TEST PLAN 4: REMOVE SENSOR DIALOG FRAGMENT	70
7.4.1 <i>Summary</i>	70
7.4.2 <i>Test Cases</i>	70
7.5 TEST PLAN 5: SEATING ACTIVITY.....	73
7.5.1 <i>Summary</i>	73
7.5.2 <i>Test Cases</i>	73
SPRINT 2	74
7.6 TEST PLAN 6: APP LAYOUT	74
7.6.1 <i>Summary</i>	74
7.6.2 <i>Test Cases</i>	74
7.7 TEST PLAN 7: ADMIN ROOMS	75
7.7.1 <i>Summary</i>	75
7.7.2 <i>Test Cases</i>	75
7.8.1 <i>Summary</i>	77
7.8.2 <i>Test Cases</i>	77
7.9 TEST PLAN 9: CONNECTION.....	79
7.9.1 <i>Summary</i>	79

<i>7.9.2 Test Cases.....</i>	79
SPRINT 3	80
7.10 TEST PLAN 10: NETWORK CONFIGURATION	80
<i>7.10.1 Summary</i>	<i>80</i>
<i>7.10.2 Test Cases.....</i>	<i>80</i>
7.11 TEST PLAN 11: ADMIN/USER	81
<i>7.11.1 Summary</i>	<i>81</i>
<i>7.11.2 Test Cases.....</i>	<i>81</i>
7.12 TEST PLAN 12: LOGIN INDICATOR.....	82
<i>7.12.1 Summary</i>	<i>82</i>
<i>7.12.2 Test Cases.....</i>	<i>82</i>
7.13 TEST PLAN 13: LOG FEATURE FOR AN ADMIN USER.....	83
<i>7.13.1 Summary</i>	<i>83</i>
<i>7.13.2 Test Cases.....</i>	<i>83</i>
VIII. ETHICAL DIMENSIONS	84
8.1 INDIVIDUAL AND SOCIETY	84
8.2 STAKEHOLDERS	84
8.3 POTENTIAL ETHICAL CONSIDERATIONS.....	85
IX. COMPUTER SIMULATION.....	86
X. PRODUCT PITCH	87
XI. FINAL ORAL PRESENTATION SLIDES.....	89
XII. PRODUCT APP.....	105
XIII. FINAL TEAM BLOG	107
XIV. EXPECTATION OF ORIGINALITY FORM	110

INTRODUCTION

The product that our team has developed is a seat tracking app that allows users to find a place to sit on their android app. The reason as to why we decided to make this app was to help people save time looking for a seat that is tailored to their needs.

The app is designed to track seats in real time with the help of load cell sensors that are mounted to a unique seat in a particular area. This sensor generates a signal to a cloud and notifies the app about the seat's availability. But before being able to access the app, the user is required to sign up to the app by using their google account which will be stored in a separate cloud. Our objective is to design an app that is helpful, practical, and easy to navigate for users so that they can go find a seat without any hassle.

The product will be especially useful for university students as well as the general populace who are looking for a place to study peacefully, relax, or even socialize.

I. MISSION STATEMENT

This section outlines the mission statement for our product: The iSeat. This includes the product description, benefit proposition, key business goals, target market, assumptions, constraints, and stakeholders.

Product Description:

The Seat Tracker App monitors seats in commonly used study rooms in order to provide users with real-time availability of seating in the room. The seat monitor consists of a pressure sensor connected to a Wi-Fi module, which can send the current seat status (empty/occupied) into a database. The user will then be able to scroll a list of rooms and view the number of seats available in each on a smartphone application interfacing with the database.

Benefit Proposition:

The product will offer real-time updates on seat availability to help students find seating in the school's library, computer labs, and other areas in a more efficient manner. This will help students start studying quicker, save time, and maximize their study sessions while using school facilities. Students will no longer waste time looking for an empty seat, instead they will be able to consult the app and immediately be directed to an available seat.

Key Business Goals:

Per unit costs: The load sensors are \$4.50 and save \$0.50 per 100 units. As for the WiFi modules they cost \$1.40 per unit. This would mean in order to be able to support these projects we would need to charge 10\$ per unit.

Development Team Cost: Team members will be spending 20h/week developing the product at 30\$/h for a total of 6 weeks and 6 team members. The total cost for the development team will be around \$21,600.00.

Estimate for number of units: Scholarly institutions generally have more than one study room, with many seats for each room. At Concordia Library for example, that's ~170 seats per room, 3 rooms per floor, with 5 floors, for a total of around 2,550 seats needing a sensor.

Profitability: $10 * (\text{units}) - 5.90 * (\text{units}) - 21,600 = 0$

$$\text{Units} = 5268.293$$

Taking into account the unit price and cost, as well as development costs, ~5269 units need to be sold before our product starts to make profit. With a projected 2,550 seats for one single school, we will have to find 2-3 schools willing to invest in our product in order for it to become profitable. Taking into consideration that this is a start-up project we believe that we can attain profitability within a year's time.

Target Market:

The product will be targeted towards higher education institutions that want to provide their students an easier and more streamlined study environment.

Assumptions:

- 1) Students have a hard time finding available seats to study at school.

Validation: TRUE. Interviews revealed this was a problem for students.

- 2) The sensor can handle a load of at least 40 kg (i.e., a person).

Validation: FALSE/TRUE. The Film Sensor was unable to detect weight accurately/reliably, while the Load Cell Sensor was able to detect and output weights up to ~55 kg. We therefore decided to use the Load Cell sensor in the product.

- 3) There will be no problem sending data over the school Wi-Fi to a database.

Validation: FALSE. In order to test and Demo at school we need to create a Mobile Hotspot from a cell phone to connect the Arduino to the internet.

- 4) Sensors will be able to accurately detect when someone is sitting vs empty seat.

Validation: TRUE.

- 5) The selected seats will be able to accommodate a sensor.

Validation: FALSE. We ended up building our own seat-sensor prototype for the product.

Constraints:

- 1) The project must be completed within the timeframe: November 28 to December 4.
- 2) The project must include a sensor interfacing with smartphone app.
- 3) The app must be able to track the number of seats available in real-time.
- 4) The network bandwidth used for the app is limited by the school's Wi-Fi.
- 5) Seating comfort is not diminished once the sensor is installed to a seat.

Stakeholders:

- 1) School Management (Buying Customer / Investors / Chair sensor installers)
- 2) Student Body (Using Customer)

II. FINAL PRODUCT BACKLOG

Sprint 1 PB:

Product Backlog								
Story ID	Title	Card	Story Points	Sprint	Status	Priority	Conversation	Confirmation
Sprint 1 (PB v1)								
HW-1	Hardware Setup	As a customer, I want a sensor device that can be placed on seats so that I can track its availability for people to use.	20	Sprint 1	DONE	High	Have to hook up the sensor to the uC and then find a way to send data to the android app. First test sensor operation with uC. Then test sending some data from uC to app.	uC detects (show light or smtg) when sensor is pressed. Data sent from uC appears on app.
DISC-1	Arduino Experiment/Practice	As the developer, I want to practice working with the Arduino uC and the sensors so that I know what to do for the product.	20	Sprint 1	DONE	Medium	Mock around with the Ardunio hardware, connect the Load Cell and the Film pressure sensor and learn how they behave. Practice with the WiFi Module, hook it up to the router and send some HTTP requests. Find out calibrations/circuit components needed to make sensors work.	
SOF 1	Add Sensor Dialog Fragment	As an admin, I want to see a list of assigned seat schools I can click on to add my sensor to a room I'm managing.	8	Sprint 1	DONE	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Create a dialog fragment that takes as input the roomKey for sensors to be added. Display info on the selected room. List sensors that are connected to Firebase but not assigned to a room. When sensor is clicked, open a confirmation alert dialog. Confirming will set the sensor roomID value in Firebase to the roomKey of the room. Show no available sensors if none available.	+ Displays the correct room info + Shows no sensors message when none are available + List is updated when a sensor is assigned to room + When sensor is clicked, open a confirmation alert dialog + List is updated when a sensor becomes available
SOF-2	Remove Sensor Dialog Fragment	As an admin, I want to see a clickable list of seat sensors currently assigned to a room so that I can remove it and re-use it in another room.	8	Sprint 1	DONE	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Create a dialog fragment that takes as input the roomKey so it can find sensors in that room. Display info on the selected room. List sensors that are assigned to that room on Firebase. When sensor is clicked, open a confirmation alert dialog. Confirming will set the sensor roomID value in Firebase to 0 (removing it from the room). Show no sensors message if room has none.	+ Displays the correct room info + Shows no sensors message when none are assigned to room + List is updated when the sensor is removed from room
ACT-2	Profile Activity	As a student, I want to have a login/sign up page	40	Sprint 1	DONE	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Need to create a database to store all login information. Profiles also need to be able to save their preferences and should link to their correct database.	Profile data entered is stored to database. Name, Password
ACT-1	Seating Activity	As a student, I want to view a list of study rooms with available seats so that I waste less time looking for seats when I could be studying.	40	Sprint 1	DONE	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Need to plan the layout.xml file for adding profiles student & admins. Write java to handle activity layout. OB for profile data.	List of rooms with available seats displayed on screen.
SIM-1	Computer Simulation Plan	As the developer, I want to simulate the product to validate some assumptions that were made about the product	20	Sprint 1	DONE	High	Discover the minimum reliable configuration for the sensors on seats. 1 sensor be enough? Where should we place it? Consider 2x2 of Seats? 3 sensors in circular pattern? 4 sensors in square pattern? etc. Create 3-D surface plots of seats, sensors, and students in software (SolidWorks or similar). Test contact/overlap between student and sensor(s) on seat using various seat sizes, sensor positions, and student shapes.	The discovery work is complete
SIM-1.1	Seat 3D Surface Plots	As a dev, I want to make 3d Seat plots for my product simulation	5	Sprint 1	DONE	High	Use matlab/modelem software to create a couple models of different seat shapes. 3-D models could probably be found online. Not sure how to export seat surface as (x, y, z) surface plot. Load (x, y, z) surface plot into Matlab.	The discovery work is complete
SIM-1.2	Sensor 3D Surface Plots	As a dev, I want to make 3d sensor plots for my product simulation	5	Sprint 1	DONE	High	Use matlab/modelem software to create multiple models of different sensor configurations. Find (x, y, z) plot of a pressure sensor online, or create your own from scratch (dimension in its description and build your own model). Then think about different configurations with the sensor: 1 in middle, 3 in circle (different radiusses), 4 in square (different side lengths?). This part we can go crazy, create as many different models as we want, and test them all for the best. Load (x, y, z) surface plot into Matlab.	The discovery work is complete
SIM-1.3	Student 3D Surface Plots	As a dev, I want to make 3d student plots for my product simulation	5	Sprint 1	DONE	High	Use matlab/modelem software to create multiple models of a student sitting on a seat (big, medium, small person). This model is complicated, and we could use Auto-Ergo-Master for some advice. On the other hand, 3D models of the human body are probably easy to find online, but we wouldn't need full body models, just the general area that comes in contact with the seat. Load these surface plot models into Matlab.	The discovery work is complete
SIM-1.4	Perform Simulation	As a developer, I want to perform the simulation and obtain results that will help me build my product.	3	Sprint 1	DONE	High	Use the 3-D surface plots of seats, sensor configuration, and students and run many simulations with different combinations of each. Determine which Sensor Configuration offers the MOST ROI/ABR contact between student and sensors among all seat and student shapes.	The discovery work is complete
DES-1.1	Design Prototype Seat	As a customer, I want to get an idea on what the final product will look like	8	Sprint 1	DONE	Low	Have to find hardware and plywood to create frame of chair. Use rubber eraser as a cushion for spring mechanism. Create pressure point underneath seat of prototype to activate seat sensor.	Display data when weight is applied to the seat. Also make sure the applied pressure is gone once mass is removed
DES-1.2	Design Sensor Frame	As an admin, I want the sensor to stay in place so that I do not have to keep fixing/adjusting the sensor's position when it is being used	8	Sprint 1	DONE	Low	3D print plastic frame for sensor so that it can stay in place.	Position of sensor is kept in place when the product is in use.
Velocity of Sprint 1							190	
Total Points of Sprint 1							190	100.00%

Sprint 2 PB:

Story ID	Title	Card	Story Points	Sprint	Status	Priority	Conversation	Confirmation			
Sprint 2 (PB v2)											
ACT-3	Setting Activity	As a user, I want to be able to edit my profile information (username/password), and be able to logout from the app, I also want to have the option to delete my profile at any point if I no longer need to use the app	40	Sprint 2	DONE	High	Create Setting Activity that holds user information (username/password), give the ability to edit those information and include a delete profile option (should ask for a confirmation before deleting)	When navigating settings toggle buttons should work as intended and text edits should	ACT-3	40	DONE
CON-1	Sensor Connection Status	As an administrator, I want to see the sensors' connection status so that I can confirm that the product is working properly.	8	Sprint 2	In progress	High	The administrator should be able to see the connection status of each sensor in the system. This will let the administrator know if the product is working properly. Maybe by having a notification of the connection status of the sensor on the app.	The administrator can see the connection status of each sensor in the system via a list.	CON-1	8	In progress
AL-1	App Layout	As a customer, I want to enjoy using the website and feel the consistency in the app design	20	Sprint 2	DONE	High	Create a theme for all activities, using a unified font family and a color palette derived from the app logo, and update all the xml files accordingly	App layout is now consistent with a fresh new design and colors that are enjoyable and user-friendly.	AL-1	20	DONE
PU-2.1	Profile User	As a user, I want to be able to save and access a room into a list of favorite rooms.	8	Sprint 2	In progress	Medium	Add a clickable drawable next to each room that saves it into a list where you can access later	When user clicks favorite button drawable should change into a gold color	PU-2.1	8	In progress
PU-2.2	Profile User	As a user I want to have a list of available rooms.	4	Sprint 2	In progress	Medium	Add a clickable list that shows available rooms to user	When clicking on list there should be a recycler view of all the rooms and their availability	PU-2.2	4	In progress
PA-3.1	Admin Seating Activity	As a manager, I want to be able to create an admin profile to store information for the study rooms I am managing.	8	Sprint 2	In progress	High	Create db table for seats, column for associated Room ID.	When clicking on button list of rooms with their associated id's should be displayed	PA-3.1	8	In progress
PA-3.2	Deleting Sensor/Room	As an admin I want to delete a sensor from a room	4	Sprint 2	DONE	Medium	Add button to delete seats, but only for Admin use.	Admin can delete a seat and it disappears from the app	PA-3.2	4	DONE
PA-3.3	Adding Sensor/Room	As an admin, I want to be able to add/delete a seat to a room so that students can use it.	4	Sprint 2	DONE	Medium	Add buttons to add but only for admin use	Admin can add a seat and see it on the app.	PA-3.3	4	DONE
PA-3.4	Testing with more sensors	As a developer I want to make sure the app is correctly adding and deleting sensors	4	Sprint 2	In progress	Medium	Buy more sensors and create more rooms in the firebase to test multiple sensors at once	Each sensor individually displays their corresponding status on firebase	PA-3.4	4	In progress
ACT-1.6	Admin/User	As a developer I want to know when an admin signs in and when a user signs in	2	Sprint 2	In progress	Low	Create two separate xml pages corresponding to the type of user	When user signs in they are directed to the user page and when the admin signs in they are directed to the admin page	ACT-1.6	2	In progress
ACT-1.7	Logged in indicator	As a user I want to have an indication of when I'm signed in	4	Sprint 2	In Progress	Medium	Create icon to indicate that the user is logged in	when pressing on it the user can see their name, go to the Setting Activity, and logout	ACT-1.7	4	In Progress
HW-2	Hardware Setup	As a customer, I want a sensor device that can be placed on seats so that I can track its availability for people to use.	16	Sprint 2	DONE	High	Have to hookup the sensor to the uC and then find a way to send data to the android app. First test sensor operation with uC. Then test sending some data from uC to the app.	uC detects (show light) when the sensor is pressed. Data sent from uC appears on the app.	HW-2	16	DONE
DF-1.1	Fix latency bug	As a developer I want to make sure that there is little to no latency when redirecting from register activity to main activity	4	Sprint 2	In progress	Medium	Make a more efficient method that can communicate with firebase from the hardware	When wifi status is at its highest negative gain dB latency is at its lowest	DF-1.1	4	In progress
DF-1.2	Layout fixing	As a developer I want to make sure that the layout of screen is consistent between different platforms	4	Sprint 2	In progress	Medium	Make sure constraints on each xml page are adjusted	When layout is consistent on every device	DF-1.2	4	In progress
TEK-1	Technical work	As a developer I want to make sure that the documentation for the current and next sprint are up to date	6	Sprint 2	DONE	Medium	Groom backlog, add ghi's, and write report for submission	No tasks left to do	TEK-1	6	DONE
							Velocity of Sprint 2	98			
							Total Points of Sprint 2	136			72.06%

Sprint 3 PB:

Story ID	Title	Card	Story Points	Sprint	Status	Priority	Conversation	Confirmation			
Sprint 3 (PB v3)											
ACT-1.7	Admin/User	As a developer I want to know when an admin sign's in and when a user sign's in	2	Sprint 3	DONE	Low	Create two separate .xml pages corresponding to the type of user	When user signs in they are directed to the user page and when the admin signs in they are directed to the admin page	ACT-1.7	2	DONE
ACT-1.8	Logged in indicator	As a developer I want to know when an admin sign's in and when a user sign's in	4	Sprint 2	DONE	Medium	Create icon to indicate that the user is logged in	when pressing on it the user can see their name, go to the Setting Activity, and logout	ACT-1.8	4	DONE
ACT-1.9	QR Code	As an Admin, I want to be able to input my admin code using a QR code/Barcode scanner so that it is easier to create an admin profile.	4	Sprint 3	DONE	Medium	Admins will have the option to either sign up using the built in QR reader or by inputting the admin code manually.	Enabling the devices onboard camera and QR reading functionality to allow scanning of admin codes.	ACT-1.9	4	DONE
ACT-3.2	Delete profile option	As a user, I want to be able to delete my profile, access my profile settings and set/edit my username	4	Sprint 3	DONE	Medium	Create a clickable icon which could direct the user to a popup where they will be able to have the option to delete their account	The popup will confirm if the user would like to delete their account and will send the user to the welcome page	ACT-3.2	4	DONE
ACT-3.3	Settings activity	As a user I want to access my profile settings	4	Sprint 3	DONE	Medium	Create a clickable from the toolbar drop down or icon that will direct user to a page where they can change certain preferences	When pressing the icon, the user will be directed to the preference page which will be indicated on the toolbar	ACT-3.3	4	DONE
ACT-3.4	Username	As a user, I want to be able to give myself a username	2	Sprint 3	DONE	Low	Create a textbox in my profile page where i can write my personal username	Add an edit button that will allow user to give access to the textbox where he can change and save it	ACT-3.4	2	DONE
ACT-3.5	QR Code	As an Admin, I want to be able to input my admin code using a QR code/Barcode scanner so that it is easier to create an admin profile.	4	Sprint 3	DONE	Medium	Admins will have the option to either sign up using the built in QR reader or by inputting the admin code manually.	Enabling the devices onboard camera and QR reading functionality to allow scanning of admin codes.	ACT-3.5	4	DONE
PA-4.1	Create Admin Profile	As a manager, I want to be able to create an admin profile to store information for the study rooms I am managing.	8	Sprint 3	In Progress	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Create db entry for admin profile, add columns for rooms, info like location and so on.	Admin data entered is saved to db. Can sign-in using existing Admin Profile.	PA-4.1	8	In Progress
PA-4.2	Edit seats	As an admin, I want to be able to add/delete a seat to a room so that students can use it.	8	Sprint 3	In Progress	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Create db table for seats, columns for associated Room ID. Add buttons to add or delete seats, but only for Admin use.	Admin can add a seat and see it on the app. Admin can delete a seat and it disappears from the app	PA-4.2	8	In Progress
PA-4.3	LOG list feature for an Admin user	As an admin I want to have access to a list of users that have created accounts so that I can track how many accounts are existing	8	Sprint 3	DONE	High	Create a recycler View list where the admin will be able to scroll through a list of users	Can see a list of users which have been created which can show the date it was created, their username, their role and email.	PA-4.3	8	DONE
PA-1.1	Create Student Profile	As a student, I want to be able to create a profile so that my preferences are saved.	8	Sprint 3	DONE	High	*SEE SKETCH SHEET FOR ACTIVITY SKETCHES* Create db for student profiles type, or general DB for Admin+Student with a Type column. Students and Admins will have different access to app features.	Data entered is stored in DB as student profile. Can use existing profile to sign into site.	PA-1.1	8	DONE
PA-2	Student Friends List	As a student I'd like to have a friend's list so that I study in a group when they are available.	13	Sprint 3	In Progress	Medium	Create an add button for friends and availability indicator if a friend has the app running in background	When friend is added, it stores the friend's data into database. As for the indicator when the app is running on a friend's phone, the indicator shows green, otherwise it goes grey.	PA-2	13	In Progress
FILT-1.1	Room Filter for Students	As a student, I want to be able to filter/sort the list of study rooms to find one that fits my needs.	6	Sprint 3	In Progress	Medium	Rooms need to have a database entries with certain properties (ie: N. of seats) to sort/filter	Order of preference should be a range of practical choices	FILT-1.1	6	In Progress
FILT-1.2	Room Filter by Available Devices	As a student, I want to filter/sort to classrooms with chalkboards or whiteboards so that I can help myself or others during study sessions.	6	Sprint 3	In Progress	Low	Add true or false statement for each room in the database to indicate if there is a whiteboard/chalkboard	When admin is adding a new room, a selector will be available to indicate if the room has a board	FILT-1.2	6	In Progress
FILT-1.3	Room Filter for Admin	As an admin, I want to be able to filter/sort the list of study rooms I am managing to be able to find specific rooms faster.	6	Sprint 3	In Progress	Low	Add filtering/sorting button that can toggle the study rooms number in an ascending and descending order	When admin is toggling the filter an arrow drawable flips horizontally	FILT-1.3	6	In Progress
PU-2.1	Profile User	As a user, I want to be able to save and remove a room into a list of favorite rooms.	8	Sprint 3	In Progress	Medium	Add a clickable drawable next to each room that saves it into a list where you can access later	When user clicks favorite button drawable should change into a gold color	PU-2.1	8	In Progress
PU-2.2	Profile User	As a user I want to have a list of my favorite rooms.	4	Sprint 3	In Progress	Medium	Add a clickable list that shows available rooms to user	When clicking on list there should be a recycler view of all the rooms and their availability	PU-2.2	4	In Progress
CON-2.1	Interrupt bugs	As an admin, I would like to fix any interruptions in the Load Cell so that it may accurately display its info to the firebase	8	Sprint 3	DONE	High	Add a watchdog timer for when the arduino has lost its connection with the firebase or the sensor and have it self reset.	When sensor is disconnected, the arduino will show that it will execute a reset	CON-2.1	8	DONE
CON-2.2	Sensor Connection Status	As an administrator, I want to see the sensors' connection status so that I can confirm that the product is working properly.	8	Sprint 3	DONE	High	The administrator should be able to see the connection status of each sensor in the system. This will let the administrator know if the product is working properly. Maybe by having a notification of the connection status of the sensor on the app.	The administrator can see the connection status of each sensor in the system via a list.	CON-2.2	8	DONE
CON-2.3	More Dynamic WiFi Connection Method	As a user, I want to be able to configure the Seat sensor to connect to any wifi Network	13	Sprint 3	DONE	HIGH	Gotta make the arduino serve up a small webpage, where a network SSID and password can be entered so the user can choose a WiFi network to connect the sensor to.	The user can enter a network SSID and Password and the uC connects to it	CON-2.3	13	DONE
CON-2.4	Sensor Identification	As an admin, I want to be able to set a name for my sensor so that I can easily identify it within the app when searching for available sensors.	4	Sprint 3	In Progress	Low	Look into a simple way of having the admin set a name for the sensor, maybe a small card with the name that the admin can put in the seat	An id for the sensor can be entered on same page as network Setup. The same ID appears in the APP when looking at available sensors	CON-2.4	4	In Progress
DOC-1	Final Documentation / Project Wrap Up	As a developer I want to present my product to an important stake holder	20	Sprint 3	DONE	High	This includes preparing the presentation, final demo, final report, pitch video, sprint3 wrap-up, etc	The stakeholders (professors) should be satisfied and give feedback on the final product. The final report and other documents are submitted	DOC-1	20	DONE

Velocity of Sprint 3 93
Total Points of Sprint 3 152 61.18%

III. DESIGN DOCUMENT

The following section contains the design documentation for the iSeat product as of Sprint 1. This includes Android Application Wireframes, the System, Hardware, and Software Architectures, and finally Use Cases and Sequence Diagrams.

4.1 Android Application Wireframes

The following *Figure 1* shows a Wireframe sketch of the iSeat application, which provides a general idea of how the app will look and behave. It includes things like text, buttons, the general page layout, and navigation between one page and the next. Since the app will be constantly evolving, the sketches are more of a starting point to give inspiration when developing a given feature. Following *Figure 1* are short descriptions of each app element present in the diagram.

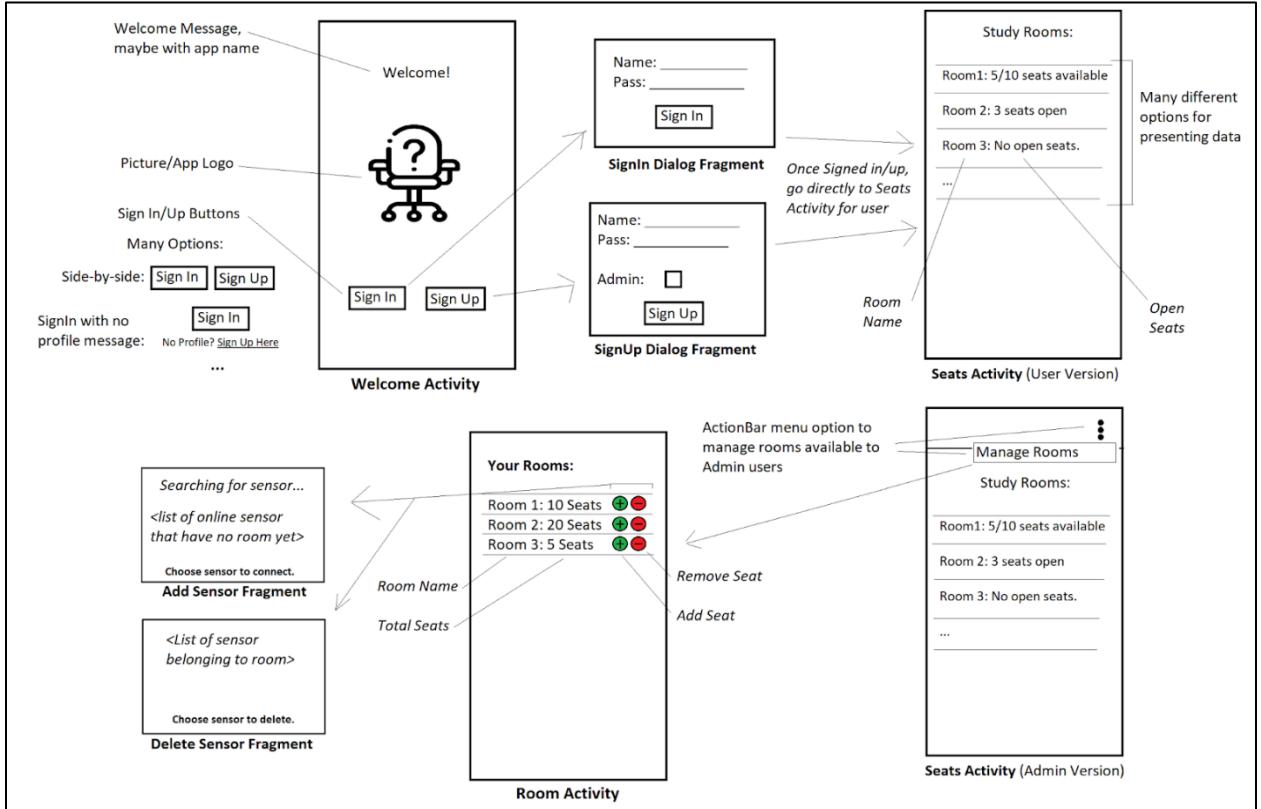


Figure 1: Wireframe sketch of app pages and navigation.

Welcome Activity: The welcome activity is the first page the user sees. It has product info like the app name, a welcome message or slogan, and the product logo. Also included are buttons to sign in or sign up, which open up either the Sign In Activity or the Sign Up Activity. If the user has signed in previously and restarted the app, they could be directed immediately to the Seats Activity.

Sign In Dialog Fragment: The sign in dialog fragment is for returning users to log in and use their saved personal preferences. Included are fields for username (email) and password and clicking the Sign In button will open up the user's Seats Activity for the appropriate type of user (admin/student).

Sign Up Dialog Fragment: The sign-up dialog fragment is for new users to create an account and save personal preferences for the app. This includes at a minimum fields for their name (email) and password. As of now, checking a box sets the account as Admin, who can add/remove rooms and their seats. Once signed Up, they are immediately signed in and the Seats Activity opens for the respective user type (admin/student).

Seats Activity - User Version: The user version of the Seats Activity displays a list of rooms set up by Admin users and shows the current occupancy of seats in the room. The user will be able to scroll through and decide which room they want to go to. Seat data for a room updates in real-time, and eventually filters and favorites features can be implemented to sort the list.

Seats Activity - Admin Version: Exactly like the User Version, except the Admin Seats Activity has a button to open up the Room management Activity. This could be located in the Action Bar at the top of the screen. Admins have this added option to add/remove rooms and add/remove the sensors in rooms.

Room Activity: The Room Activity shows the rooms that the current Admin user is managing on the app. The admins have the options to add more or delete rooms, which will appear in the seat's activity to all users. For every room, the admin will have the option of adding or removing its sensors through buttons that will either open the Add Sensor or Remove Sensor Dialog Fragments.

Add Sensor Dialog Fragment: The Add Sensor Dialog Fragment opens for a specific room and displays a list of sensors connected to the database that don't have any roomID yet assigned to them (roomID=0). The admin user can click on a sensor in the list, and a confirmation dialog will ask if they wish to add the selected sensor to the room. Upon confirmation, the sensor item in the database is updated with the roomID.

Remove Sensor Dialog Fragment: The Remove Sensor Dialog Fragment opens for a specific room and displays a list of sensors currently assigned to that room. The admin user can click on a sensor in the list, and a delete dialog will ask if they wish to remove the sensor from the current room. Upon confirmation, the sensor item in the database is updated to roomID=0, which removes the sensor from that room and makes it visible for the Add Sensor Dialog Fragment.

4.2 System Architecture

The following *Figure 2* shows the iSeat system architecture diagram. The system is made up of 5 elements: the load cell sensor on the chair, the Arduino Nano 33 IoT board, the Google Firebase Cloud Services, the Firebase Realtime Database, and the Client Android Application. Only one chair controller and client app are shown in the diagram, but the system will have multiple chair sensor-controllers as well as client applications interacting with the online database concurrently.

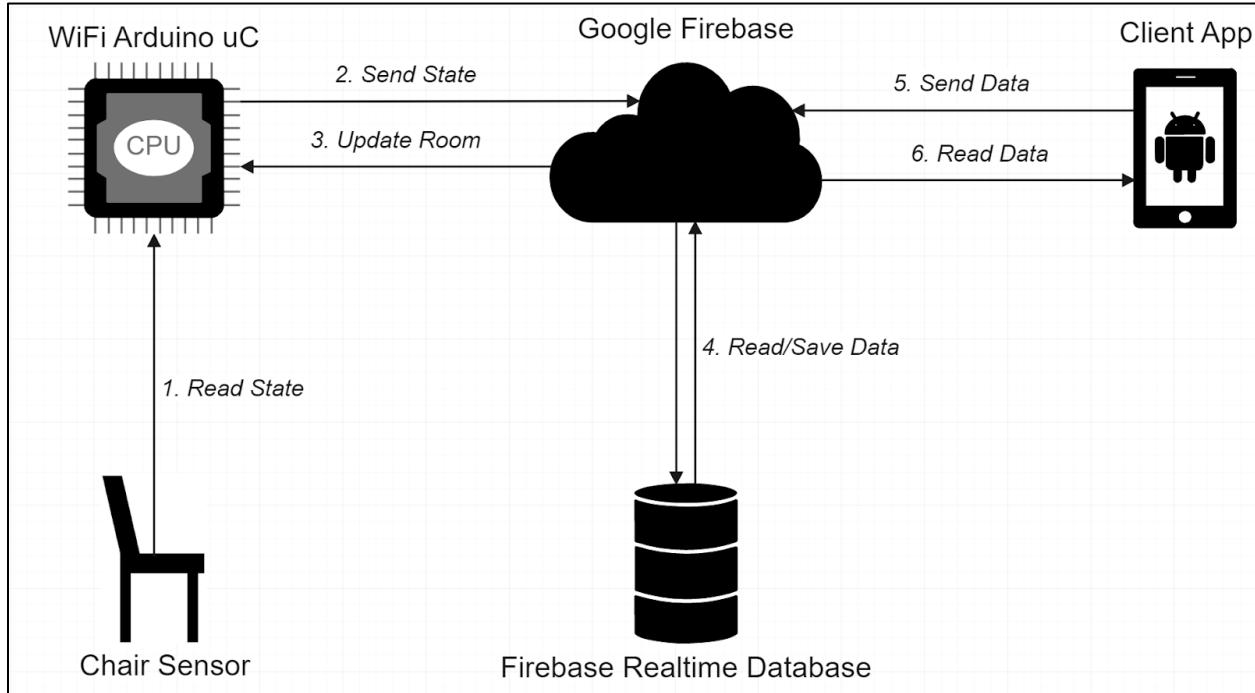


Figure 2: iSeat architecture diagram. Created using www.edrawmax.com.

Chair controller & Firebase:

1. **Read State:** uC read sensor state to determine if seat is open/taken.
2. **Send State:** uC sends sensor state to firebase.
3. **Update Room:** Firebase can update the sensor's assigned roomID.

Google Firebase & Realtime Database:

4. **Read/Save Data:** Google Firebase APIs take care of real-time data storage/retrieval.

Android Client App & Google Firebase:

5. **Send Data:** App sends data to online firebase which is accessible to all clients. This includes newrooms, profile info, and updating rooms and sensors.
6. **Read Data:** App retrieves data from firebase, like the real-time status of the sensor to show the user.

4.3 Hardware Architecture

4.3.1 Hardware

The hardware that was chosen for this project consists of a microcontroller, an amplifier, a load cell sensor, and two 1 Kilo Ohm resistors.

The load cell sensor is currently configured as a single component which measures the weight of a person at its fulcrum. It does this by measuring an internal resistance that is designed in a diamond pattern, using four 1 kilo ohm resistors, which is called a Wheatstone bridge. Two of the resistors detect compression and the other two detect tension. When you feed the Wheatstone a current, the sensor will then output a voltage depending on the force applied on the fulcrum which can be shown in the figure below.

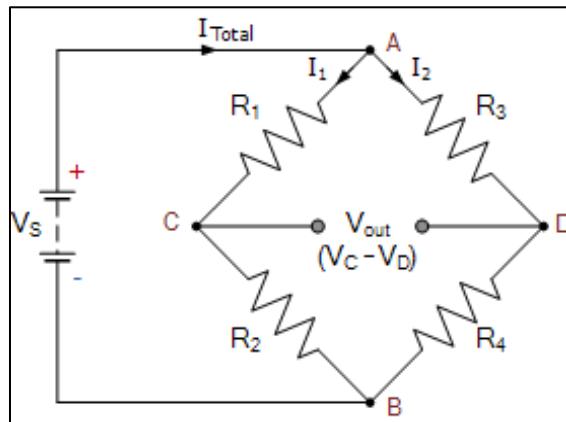


Figure 3: Load Cell Sensor Internal Configuration

However due to the loadcell's low output voltage, the microcontroller would not be able to properly pick up the change in voltage. That is why we introduce an amplifier to the system so that it can better read this change in voltage.

The chosen amplifier for this project was the HX711 load cell amplifier which is illustrated in figure 4. Due to the prototype's single sensor configuration, the current that will be fed into the Wheatstone bridge will still be too high according to its rating, hence it needed two 1 kilo Ohm resistors in series at the amplifier's analog input. Another issue that is encountered is the low output voltage from the sensor. Once the sensor's voltage has been read and converted, it is then fed into the microcontroller. Although we could have used a simple amplifier configuration to increase the output voltage of the sensor, it would not have been enough to properly read the sensor's data since we would have needed to convert the analog data into digital which would have been cumbersome. Instead, we opted into using the ADC amplifier which does both amplification and conversion at the same time.

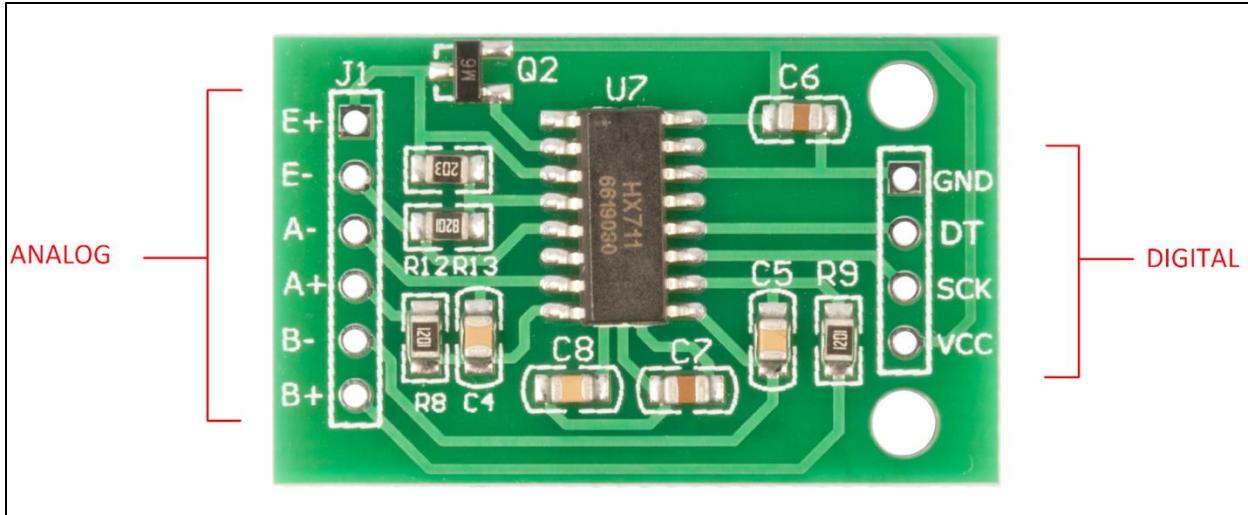


Figure 4: HX711 Amplifier & Analog to Digital Converter

When we've reached the microcontroller and supplied it with a minimum of 3.85 volts in the Vin, the CPU reads and processes the data from the ADC amplifier through the digital pins 5 and 4 where it can then provide the data to the Arduino code and distribute the information to the user via a cloud.

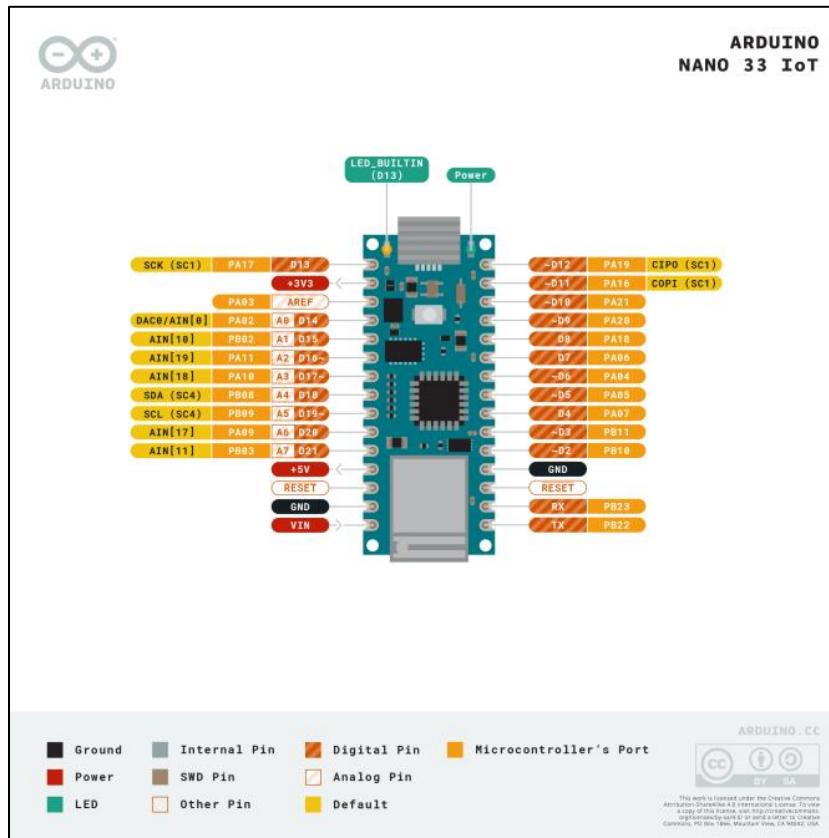


Figure 5: Arduino Nano IoT 33 Schematic

4.3.2 Prototype

The initial miniature model that was presented in sprint 1 was made using a simple piece of lumber and a small sheet of thin plywood. It was designed in the shape of a two-legged chair, with two chopped pieces of a rubber eraser as a spring to dampen the load on the sensor, and a small wooden nub made to apply pressure onto the fulcrum of the sensor, and list of illustrations of this can be shown in the figures below.

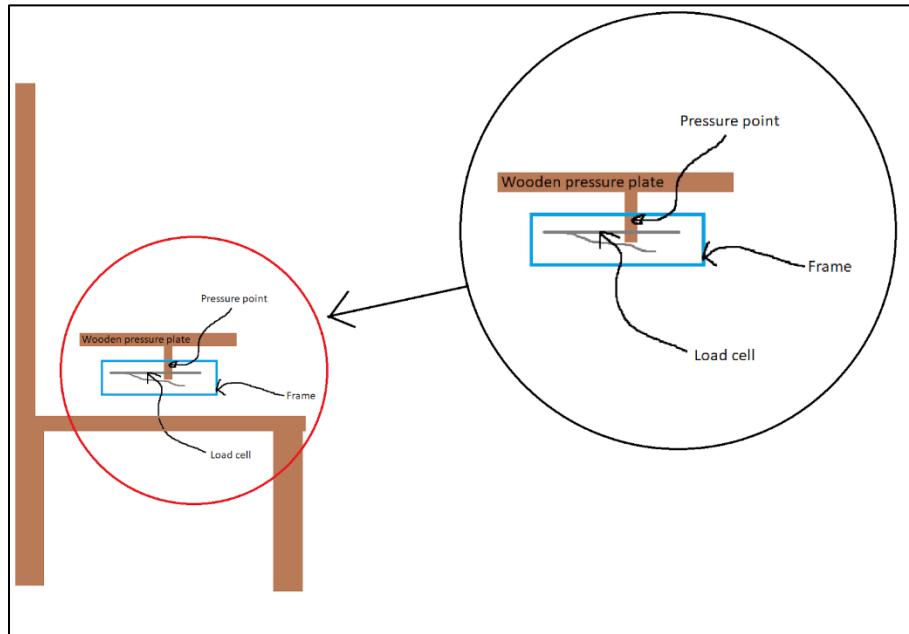


Figure 6: Illustration of Miniature Chair Model

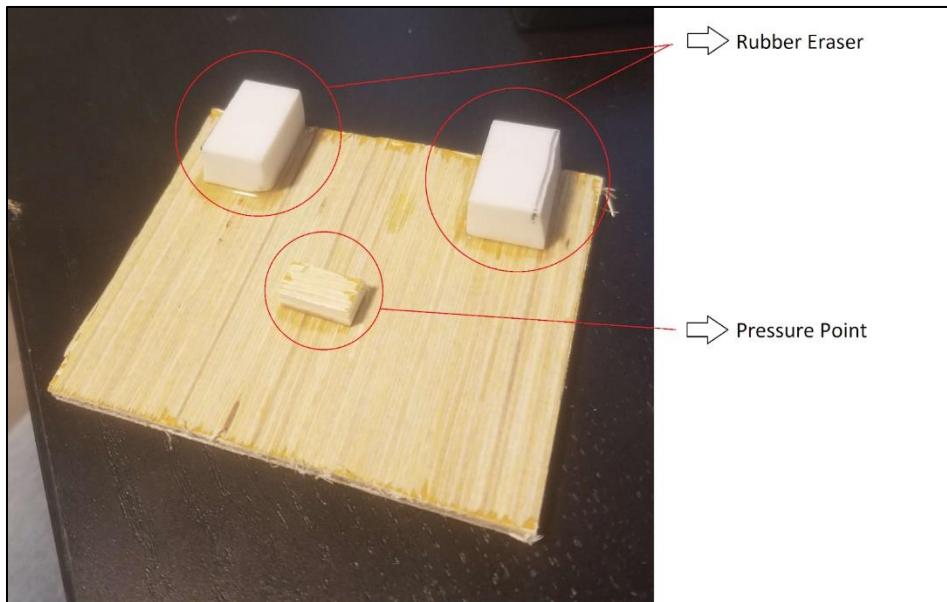


Figure 7: Close up of Seat Plate

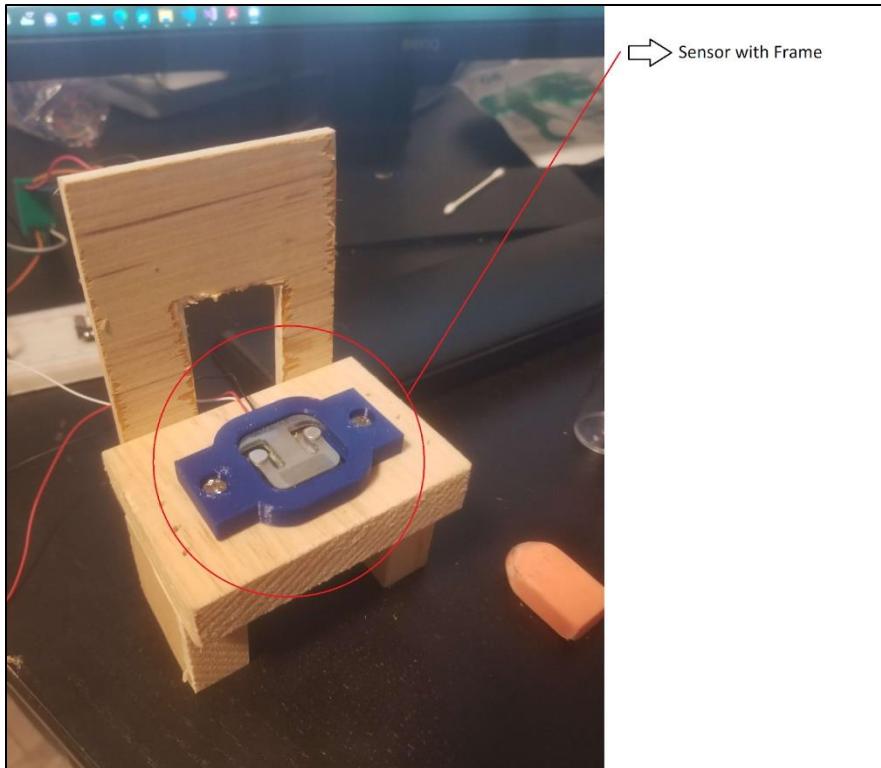


Figure 8: Chair with Framed Sensor



Figure 9: Final Miniature Model Product

The larger scale model that was made for sprint 2 consists of a hinged wooden box, an aluminum plate with a pressure point attached to it, 3 sheets of wrapping foam, a 2-inch layer of wood, and some aluminum tape.

The hinged wooden box is designed to be the base and frame for the device, it is made out of a decorative crate to carry wine. Holes were drilled into the frame so that the wires for the sensor could be threaded through.



Figure 10: Base Frame

The aluminum plate is designed to be the weight that would be applied to the sensor, it is made out of a tool designed for plastering called a "Hawk". The tool's handle was removed and replaced with a long bolt to press against the sensor's load.



Figure 11: Weight Plate with Pressure Point

The three sheets of wrapping foam were designed to be used as a way to dampen the load on the sensor, they were cut to size and glued to the underside of the aluminum plate.



Figure 12: Wrapping Foam

The 2-inch layer of wood is designed to be used as a way to increase the height of the sensor so that it can make contact with the long bolt.



Figure 13: 2" Wood Height

Finally, we glued the load cell underneath the pressure point and added some aluminum tape along the metal plate's edges to secure the load cell and the pressure point's position.



Figure 14: Finished Prototype

The larger scale model was successful in applying pressure onto the sensor, however the wrapping foam did not seem to be necessary since the metal plate provided enough resistance for the sensor.

IV. Software Architecture

This section covers the software developed for the iSeat device. This includes the code written for the Arduino used to configure its Wifi network and the main code that reads the sensor and sends updates to Firebases. Used to explain the microcontroller software is a State Machine and flowchart with the behaviour of the code.

4.4.1 Arduino Software

The Moore State Machine in Figure 15 below shows the overall behaviour of the iSeat device. Since it is an internet device, it needs to be configured with a Network SSID and password to connect to and communicate with the Google Firebase server. Furthermore, it is desired that the user can enter their custom network info. Therefore, the final version of the iSeat program is a state machine that goes through the following states: Unconnected, Connecting, and Connected. More info on each state follows the figure.

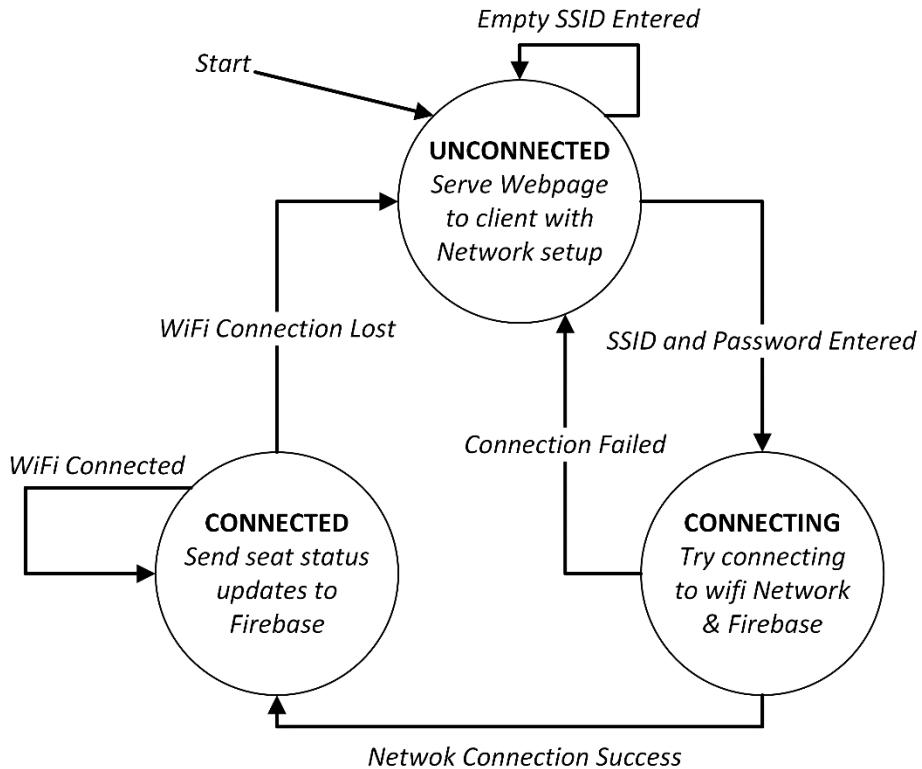


Figure 15: State Machine Transition Diagram for the state of the Arduino iSeat device.

While in Unconnected State, the device functions as a Wireless Access Point that allows internet devices to connect to it so that it can serve up a small webpage. Using either a smartphone or a wireless laptop, one can open their available Wifi networks in settings and connect to the “iSeat” access point (AP). Once connected to the AP, opening a web browser and navigating to the hostname/website <http://192.168.4.1/> will open the iSeat network configuration webpage. On this page are input boxes for a network SSID and Password to use to connect to the internet as well as a submit button. Once the user inputs their SSID and Password and clicks enter, the device shows a new page with an attempting to connect message showing the user input, and enters the Connecting State.

In the Connecting State, the device simply tries to connect to the network specified by the user’s input. It stops functioning as an Access Point and closes that connection, and the “iSeat” network disappears from the available Wifi connections of the phone/laptop used to configure the iSeat device. If it cannot connect to the given network, the attempt will time out after 10 seconds and return to the Unconnected State. The “iSeat” network will also reappear since the Access Point has been re-established. Upon revisiting the webpage, the user will see their last input and an error message describing that the device can’t connect to the network specified by that SSID and password. If the connection is successful, the device enters the Connected State and starts normal operation.

The Connected State is the normal operating mode where the iSeat device sends updates on the seat status to the Firebase server depending on the input from the seat sensor. A flowchart of the general program flow in this state can be seen in Figure 16 on the following page. The main loop of the program has a small delay at the start, followed by a check to see if it is still connected to the internet. If so, it exits the state to return to the Unconnected state where the user can re-configure its network connection. Afterwards, it uses a counter variable to perform a check every 5 seconds to see if it is currently assigned to a room and updates its “hasRoom” flag. It performs a read operation from Firebase, since the basic (unpaid) version of Firebase use for the app does not have the server-side functions feature, and we are not capable of making Firebase send data to the Arduino without a read request. Following that, it will start polling its sensor input for the current seat status only if it is assigned to a room. If it is not assigned to a room, it will blink its on-board LED at a consistant rate. When assigned to a room, the LED will indicate the current seat status, with ON for open and OFF for closed. This data will then be sent to the Firebase server, but a variable is used to keep track of the last value of the sensor on the previous poll, and so the update to Firebase will only be sent if there is a change in the sensor state.

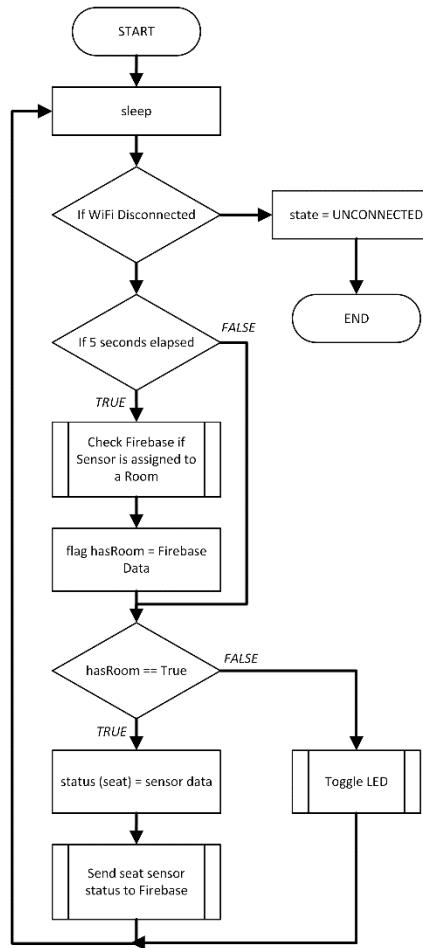
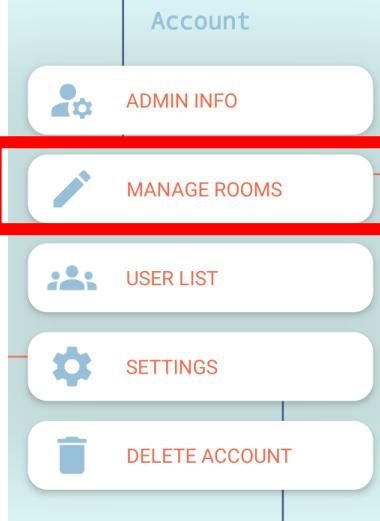
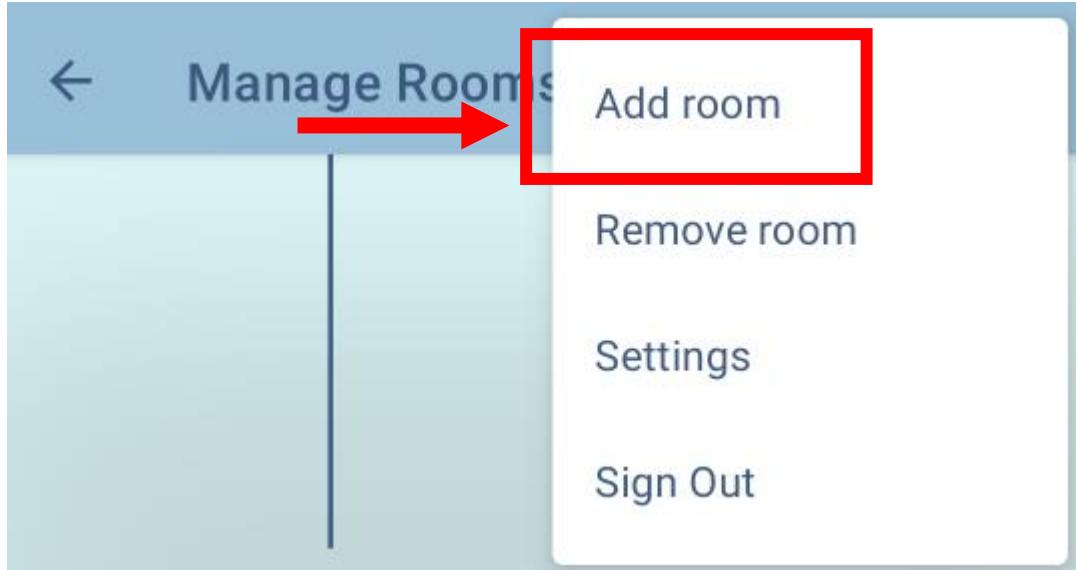


Figure 16: Flowchart representing the general program flow when the Arduino iSeat device is in Connected State (normal operation).

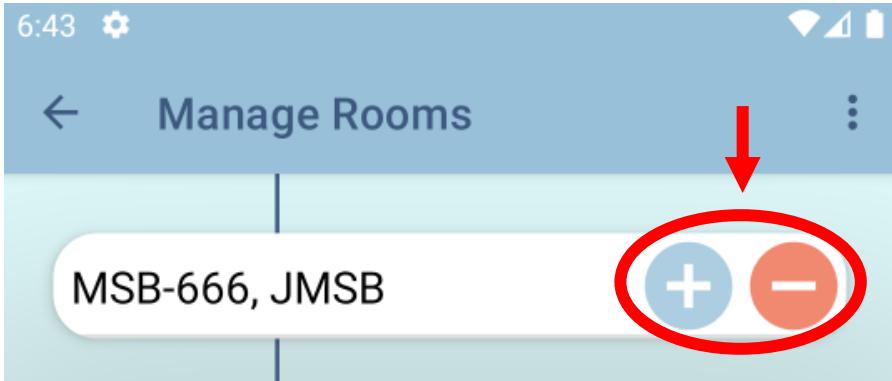
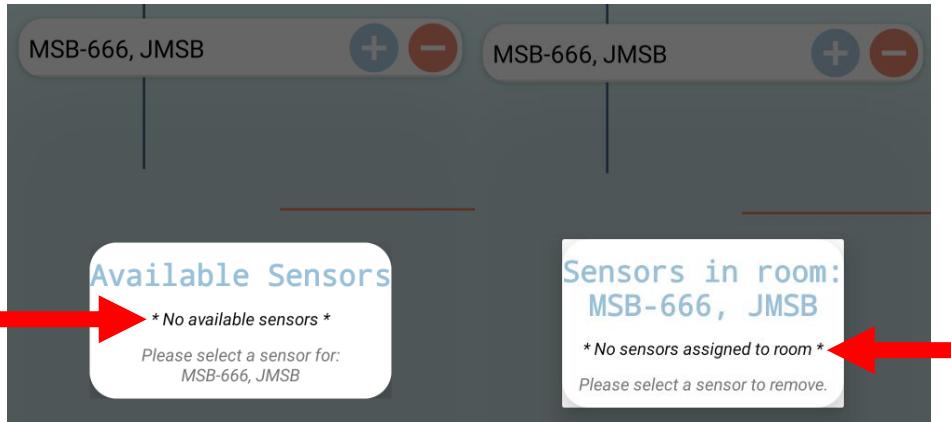
V. USER MANUAL

Admin Installation Guide for the iSeat – Seat Tracking App	
<i>1. Install the app on your mobile device</i>	
a.	Go to the Google Play Store on your device and search for “iSeat – Seat Tracking App”
b.	Download and install the app
<i>2. Prepare the device for installation</i>	
a.	Make sure the Arduino Nano IoT33 inside the seat is connected to a DC power source adapter via Micro-USB
b.	<p>Connect the device mentioned in step 2.a. to a secure Wi-Fi network by completing the following steps:</p> <ul style="list-style-type: none"> i. Make sure the LED on the Arduino nano is solid green ii. The Arduino nano will then create a network called ‘iSeat’ which you will connect to that using a phone or a laptop. iii. You will then open a web browser and write ‘http://192.168.4.1/’ in the address bar iv. Once on this page you must write your SSID and password then click submit.
<i>3. Install the seats</i>	
a.	Attach the seats on top of the existing upholstery
b.	Ensure that each seat is correctly strapped onto the chair’s underside

4. Setup the room list	
a.	Sign up on the app
b.	Once you've signed in, on the toolbar press the admin profile button on the top left of the display. 
c.	Next, click on the “Manage Rooms” button. This will direct you to the list of rooms where you can add and remove rooms. 

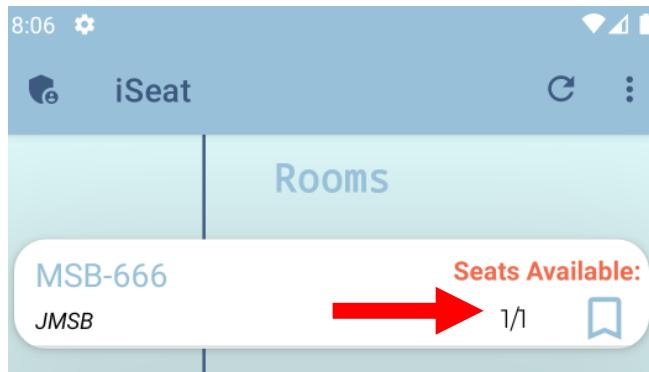
	<p>Once here, on the toolbar press the drop-down button on the top right of the display and select add room.</p> 
d.	<p>Here you will be able to add the name of the building the room is located in on the first text box and the room's number in the second text box.</p> 

5. Setup the sensors

	<p>Then you will choose to either add or remove a sensor from a list to add to the new room by pressing the add or remove button.</p>
a	
b.	<p>A window will appear where you will be able to choose a sensor to either add or remove from the chosen room.</p> 

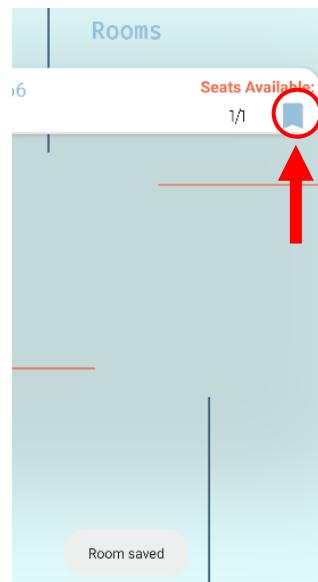
6. Testing the sensor

- a. Once you have completed steps 1 – 6, go back to the rooms page. You will then see that the room will display the number of seats that will be available out of the total number of seats in the room.

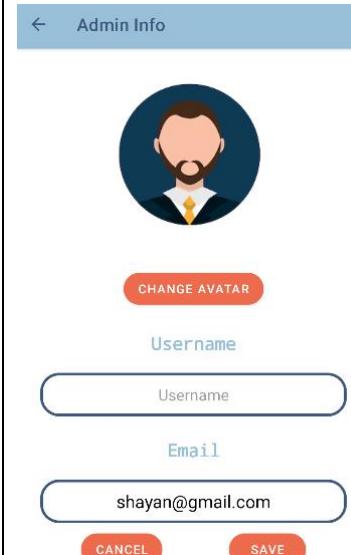


- b. You can then test the following features to verify all the functionality is working as intended.

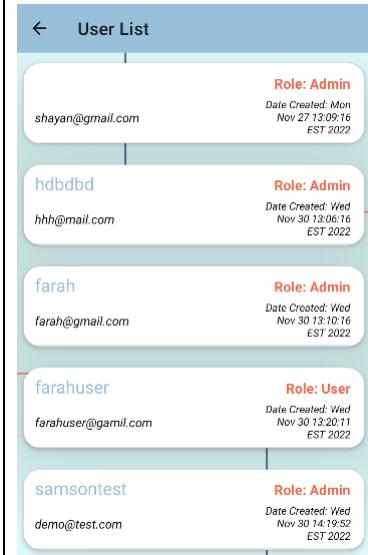
i. Saving rooms



ii. Editing username & avatar



iii. Checking user list and much more



VI. DEFINITION OF DONE

Sprint 1

Story ID: HW-1 (Hardware - 1)	
User Story:	
	<u>The Card</u>
As a customer, I want a sensor device that can be placed on seats so that I can track its availability for people to use.	
	<u>The Conversation</u>
Have to hookup the sensor to the uC and then find a way to send data to the android app. First test sensor operation with uC. Then test sending some data from uC to the app.	
	<u>The Confirmation</u>
uC detects (show light or something) when the sensor is pressed. Data sent from uC appears on the app.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done N/A N/A N/A Done
Zero Known Defects	N/A
Acceptance Tested	N/A
Live on Production Servers	Done

Story ID: DISC-1 (Discovery - 1)	
User Story: <u>The Card</u> As the developer, I want to practice working with the Arduino uC and the sensors so that I know what to do for the product.	
<u>The Conversation</u> Mess around with the Arduino hardware, connect the Load Cell and the Film pressure sensor and learn how they behave. Practice with the WiFi Module, hook it up to the router and send some HTTP requests. Find out calibrations/circuit components needed to make sensors work.	
<u>The Confirmation</u> Check if sensors are able to take input and communicate that input to the uC via the serial monitor. Check if the WiFi Module is also able to communicate with the server.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	N/A
Zero Known Defects	Done
Acceptance Tested	N/A
Live on Production Servers	N/A

Story ID: DF - 1 (Dialog Fragment 1)	
User Story:	
<u>The Card</u>	As an admin, I want to see a list of unassigned seat sensors I can click on to add my sensor to a room I'm managing.
<u>The Conversation</u>	Create a dialog fragment that takes as input the roomKey for sensors to be added. Display info on the selected room. List sensors that are connected to the firebase but not assigned to a room. When the sensor is clicked, open a confirmation alert dialog. Confirming will set the sensor roomID value in Firebase to the roomKey of the room. Show no available sensors if none available.
<u>The Confirmation</u>	-Displays the correct room info. Shows no sensor message when none are available. List is updated when a sensor is assigned to a room. List is updated when a sensor becomes available
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Not Done Not Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: DF - 2 (Dialog Fragment 2)	
User Story:	
<u>The Card</u>	As an admin, I want to see a clickable list of seat sensors currently assigned to a room so that I can remove it and re-use it in another room.
<u>The Conversation</u>	Create a dialog fragment that takes as input the roomKey so it can find sensors in that room. Display info on the selected room. List sensors that are assigned to that room on Firebase. When the sensor is clicked, open a confirmation alert dialog. Confirming will set the sensor roomID value in Firebase to 0 (removing it from the room). Show no sensor message if the room has none.
<u>The Confirmation</u>	<ul style="list-style-type: none"> - Displays the correct room info - Shows no sensors message when none are assigned to room - List is updated when the sensor is removed from room
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Not Done Not Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT - 1 (Seating Activity)	
User Story:	
<u>The Card</u> As a student, I want to view a list of study rooms with available seats so that I waste less time looking for seats when I could be studying.	
<u>The Conversation</u> Need to plan the layout.xml file for adding profiles: student & admins. Write Java to handle activity layout. DB for profile data.	
<u>The Confirmation</u> List of rooms with available seats displayed on screen.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done Not Done Not Done Done Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Not Done Not Done Not Done Done
Zero Known Defects	Done
Acceptance Tested	Not Done
Live on Production Servers	Done

Story ID: ACT - 2 (Profile Activity)	
User Story:	
	<u>The Card</u> As a student, I want to have a login/sign up page
	<u>The Conversation</u> Need to create a database to store all login information. Profiles also need to be able to save their preferences and should link to their correct database.
	<u>The Confirmation</u> Profile data entered is stored in the database. Name, Password.
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Not Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Not Done Not Done Done Done
Zero Known Defects	Done
Acceptance Tested	Not Done
Live on Production Servers	Done

Story ID: DES - 1 (Design)	
User Story:	
<u>The Card</u> As a customer, I want to get an idea on what the final product will look like	
<u>The Conversation</u> Have to find hardwood and plywood to create a frame for a chair. Use a rubber eraser as a cushion for the spring mechanism. Create a pressure point underneath the seat of the prototype to activate the seat sensor. 3D print plastic frame for sensor so that it can stay in place.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	N/A
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Not Done
Acceptance Tested	Done
Live on Production Servers	N/A

Sprint 2

Story ID: PA-3 (Admin Profile - 3)	
User Story: <u>The Card</u> As a manager, I want to be able to create an admin profile to store information for the study rooms I am managing.	
<u>The Conversation</u> Create database entries for admin profiles, add columns for rooms, info like location and so on. <u>The Confirmation</u> Admin data entered is saved to db. Can sign-in using existing Admin Profile.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: PU-2 (User Profile 2)	
User Story:	
<u>The Card</u>	
As a user, I want to be able to save and access a room into a list of favorite rooms and I want to have a list of available rooms.	
<u>The Conversation</u>	
Add a clickable drawable next to each room that saves it into a list where you can access later	
<u>The Confirmation</u>	
When user clicks favorite button drawable should change into a gold color and when clicking on list there should be a recycler view of all the rooms and their availability	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-3 (Activity 3.1)	
User Story:	
<u>The Card</u>	As a user, i want to be able to edit my profile information (username/password), and be able to logout from the app, i also want to have the option to delete my profile at any point if i no longer need to use the app
<u>The Conversation</u>	Create Setting Activity that holds user information (username/password), give the ability to edit those information and include a delete profile option (should ask for a confirmation before deleting)
<u>The Confirmation</u>	When clicking on button list of rooms with their associated id's should be displayed
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Not Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Not Done

Story ID: AL-1 (App Layout)	
User Story:	
<u>The Card</u> As a customer, I want to enjoy using the website and feel the consistency in the app design	
<u>The Conversation</u> Create a theme for all activities, using a unified font family and a color palette derived from the app logo, and update all the xml files accordingly	
<u>The Confirmation</u> App layout is now consistent with a fresh new design and colors that are enjoyable and user-friendly.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	N/A
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-1 (Activity 1)	
User Story:	
<u>The Card</u> As a student, I want to view a list of study rooms with available seats so that I waste less time looking for seats when I could be studying.	
<u>The Conversation</u> Need to plan the layout.xml file for adding profiles: student & admins. Write Java to handle activity layout. Database for profile data.	
<u>The Confirmation</u> List of rooms with available seats displayed on screen.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Not Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Not Done Not Done Not Done Not Done Not Done
Zero Known Defects	Not Done
Acceptance Tested	Not Done
Live on Production Servers	Not Done

Story ID: HW-1 (Hardware 1)	
User Story:	
<u>The Card</u>	
As a customer, I want a sensor device that can be placed on seats so that I can track its availability for people to use.	
<u>The Conversation</u>	
Have to hookup the sensor to the uC and then find a way to send data to the android app. First test sensor operation with uC. Then test sending some data from uC to the app.	
<u>The Confirmation</u>	
uC detects (show light) when the sensor is pressed. Data sent from uC appears on the app.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Not Done
Acceptance Tested	Not Done
Live on Production Servers	Done

Story ID: CON-1 (Connection Status 1)	
User Story:	
<u>The Card</u>	
As an administrator, I want to see the sensors' connection status so that I can confirm that the product is working properly.	
<u>The Conversation</u>	
The administrator should be able to see the connection status of each sensor in the system. This will let the administrator know if the product is working properly. By having a notification of the connection status of the sensor on the app.	
<u>The Confirmation</u>	
The administrator can see the connection status of each sensor in the system via a list.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done N/A
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Sprint 3

Story ID: PA-1 (Profile Admin)	
User Story: <u>The Card</u> As a student, I want to be able to create a profile so that my preferences are saved. <u>The Conversation</u> Students and Admins will have different access to app features. <u>The Confirmation</u> Data entered is stored in DB as student profile. Can use existing profile to sign into site.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: PA-2 (Profile Admin)	
User Story:	
<u>The Card</u> As a student I'd like to have a friend's list so that I study in a group when they are available.	
<u>The Conversation</u> Create an add button for friends and availability indicator if a friend has the app running in background	
DoD Checklist	Status
Design Reviewed	Not Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	NA
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA NA NA
Zero Known Defects	NA
Acceptance Tested	NA
Live on Production Servers	NA

Story ID: PA-4.1 (Profile Admin 4.1)	
User Story:	
<u>The Card</u> As a manager, I want to be able to create an admin profile to store information for the study rooms I am managing.	
<u>The Conversation</u> Create db entry for admin profile, add columns for rooms, info like location and so on.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	Not Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA Not Done Not Done
Zero Known Defects	Not Done
Acceptance Tested	Not Done
Live on Production Servers	Not Done

Story ID: PA-4.2 (Profile Admin 4.2)	
User Story:	
<u>The Card</u>	As an admin, I want to be able to edit seat names and
<u>The Conversation</u>	Create database table for seats, column for associated Room ID. Add buttons to add or delete seats, but only for Admin use.
<u>The Confirmation</u>	Admin can add a seat and see it on the app. Admin can delete a seat and it disappears from the app
DoD Checklist	Status
Design Reviewed	Not Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	Not Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA Not Done
Zero Known Defects	Not Done
Acceptance Tested	Not Done
Live on Production Servers	Not Done

Story ID: PA-4.3 (LOG list feature for an Admin user 4.3)	
User Story:	
<u>The Card</u>	As an admin I want to have access to a list of users that have created accounts so that I can track how many accounts are existing.
<u>The Conversation</u>	Create a recycler View list where the admin will be able to scroll through a list of users.
<u>The Confirmation</u>	Can see a list of users which have been created which can show the date it was created, their username, their role and email.
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: FILT-1 (Filters 1.1)	
User Story:	
<u>The Card</u> As a student, I want to be able to filter/sort the list of study rooms to find one that fits my needs.	
<u>The Conversation</u> Rooms need to have a database entries with certain properties (ie: N. of seats) to sort/filter <u>The Confirmation</u> Order of preference should be a range of practical choices	
DoD Checklist	Status
Design Reviewed	Not Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	NA
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA NA NA
Zero Known Defects	NA
Acceptance Tested	NA
Live on Production Servers	NA

Story ID: FILT-1 (Filters 1.2)	
User Story:	
<u>The Card</u> As a student, I want to filter/sort to classrooms with chalkboards or whiteboards so that I can help myself or others during study sessions.	
<u>The Conversation</u> Add true or false statement for each room in the database to indicate if there is a whiteboard/chalkboard	
DoD Checklist	Status
Design Reviewed	Not Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	NA
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA NA NA
Zero Known Defects	NA
Acceptance Tested	NA
Live on Production Servers	NA

Story ID: FILT-1 (Filters 1.3)	
User Story:	
<u>The Card</u>	As an admin, I want to be able to filter/sort the list of study rooms I am managing to be able to find specific rooms faster.
<u>The Conversation</u>	Add filtering/sorting button that can toggle the study rooms number in an ascending and descending order
<u>The Confirmation</u>	When admin is toggling the filter an arrow drawable flips horizontally
DoD Checklist	Status
Design Reviewed	Not Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Not Done
End-User Documentation Updated	NA
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA NA NA
Zero Known Defects	NA
Acceptance Tested	NA
Live on Production Servers	NA

Story ID: PU-2 (User Profile 2.1)	
User Story:	
<u>The Card</u>	
As a user, I want to be able to save and remove a room into a list of favorite rooms.	
<u>The Conversation</u>	
Add a clickable drawable next to each room that saves it into a list where you can access later	
<u>The Confirmation</u>	
When user clicks favorite button drawable should change into a gold color	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: PU-2 (User Profile 2.2)	
User Story:	
<u>The Card</u>	As a user I want to have a list of my favorite rooms.
<u>The Conversation</u>	Add a clickable list that shows available rooms to user
<u>The Confirmation</u>	When clicking on list there should be a recycler view of all the rooms and their availability
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: CON-2.1 (Interrupt bugs)	
User Story:	
<u>The Card</u>	
As an admin, I would like to fix any Interruptions in the Load Cell so that it may accurately display its info to the firebase	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: CON-2.2 (Sensor Connection Status)	
User Story:	
<u>The Card</u>	
As an administrator, I want to see the sensors' connection status so that I can confirm that the product is working properly.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done Done Done Done Done Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: CON-2.3 More Dynamic Wi-Fi Connection Method	
User Story:	
<u>The Card</u>	
As a user, I want to be able to configure the Seat sensor to connect to any Wi-Fi Network	
<u>The Conversation</u>	
To make the Arduino serve up a small webpage, where a network SSID and password can be entered so the user can choose a WIFI network to connect the sensor to. Cane creates state machine for Arduino code, with unconnected, connecting, and connected states. When connected, iSeat functionality is as before.	
<u>The Confirmation</u>	
The user can enter a network SSID and Password and the uC connects to it.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete Code Refactored Code in Standard Format Code is Commented Code is Checked in Code Inspected	Done
End-User Documentation Updated	NA
Tested Tested Unit Integration Tested Regression Tested Platform Tested Language Tested	Done Done Done NA NA
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: CON-2.4 Sensor Identification	
User Story:	<u>The Card</u>
	As an admin, I want to be able to set a name for my sensor so that I can easily identify it within the app when searching for available sensors.
	<u>The Conversation</u>
	Look into a simple way of having the admin set a name for the sensor, maybe a small card with the name that the admin can put in the seat
	<u>The Confirmation</u>
	An id for the sensor can be enter on same page as network Setup. The same ID appears in the APP when looking at available sensors
DoD Checklist	Status
Design Reviewed	
Code Complete Code Refactored Code in Standard Format Code is Commented Code is Checked in Code Inspected	
End-User Documentation Updated	
Tested Tested Unit Integration Tested Regression Tested Platform Tested Language Tested	
Zero Known Defects	
Acceptance Tested	
Live on Production Servers	

Story ID: ACT 1.6 (Activity 1.6)	
User Story:	
<p style="text-align: center;"><u>The Card</u></p> <p>As a developer I want to know when an admin sign's in and when a user sign's in</p> <p style="text-align: center;"><u>The Conversation</u></p> <p>Create two seperate .xml pages correspoding to the type of user</p> <p style="text-align: center;"><u>The Confirmation</u></p> <p>When user signs in they are directed to the user page and when the admin signs in they are directed to the admin page</p>	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT 1.7 (Activity 1.7)	
User Story:	
<u>The Card</u> As a user I want to have an indication of when I'm signed in <u>The Conversation</u> Create icon to indicate that the user is logged in <u>The Confirmation</u> when pressing on it the user can see their name, go to the Setting Activity, and logout	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-1.8 (Activity 1.8)	
User Story:	
<u>The Card</u> As an Admin, I want to be able to input my admin code using a QR code/Barcode scanner so that it is easier to create an admin profile.	
<u>The Conversation</u> Admins will have the option to either sign up using the built in QR reader or by inputting the admin code manually.	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-3.2 (Activity 3.2)	
User Story:	
<u>The Card</u> As a user, i want to be able to delete my profile, access my profile settings and set/edit my username	
<u>The Conversation</u> Create a clickable icon which could direct the user to a popup where they will be able to have the option to delete their account	
<u>The Confirmation</u> The popup will confirm if the user would like to delete their account and will send the user to the welcome page	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	NA NA NA Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-3.3 (Activity 3.3)	
User Story:	
<u>The Card</u>	As a user I want to access my profile settings
<u>The Conversation</u>	Create a clickable from the toolbar drop down or icon that will direct user to a page where they can change certain preferences
<u>The Confirmation</u>	When pressing the icon, the user will be directed to the preference page which will be indicated on the toolbar
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: ACT-3.4 (Activity 3.4)	
User Story:	
<u>The Card</u>	
As a user, I want to be able to give myself a username	
<u>The Conversation</u>	
Create a textbox in my profile page where i can write my personal username	
<u>The Confirmation</u>	
Add an edit button that will allow user to give access to the textbox where he can change and save it	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

Story ID: DOC-1 (Documentation 1.1 → 1.9)	
User Story:	
<u>The Card</u>	
As a developer I want to present my product to an important stake holder	
<u>The Conversation</u>	
This includes preparing the presentation, final demo, final report, pitch video, sprint3 wrap-up, etc	
<u>The Confirmation</u>	
The stakeholders (professors) should be satisfied and give feedback on the final product The final report and other documents are submitted"	
DoD Checklist	Status
Design Reviewed	Done
Code Complete <ul style="list-style-type: none"> • Code Refactored • Code in Standard Format • Code is Commented • Code is Checked in • Code Inspected 	Done
End-User Documentation Updated	Done
Tested <ul style="list-style-type: none"> • Tested Unit • Integration Tested • Regression Tested • Platform Tested • Language Tested 	Done Done Done Done Done
Zero Known Defects	Done
Acceptance Tested	Done
Live on Production Servers	Done

VII. TEST DOCUMENT

Sprint 1

7.1 Test Plan 1: Profile Activity

7.1.1 Summary

Story ID: ACT-2.1

Create a login page for the user where they can input their email and password that was used to register to the app and get logged in. If a new user, they can click a button that will send them to a register page for the app.

7.1.2 Test Cases

Test Case: 1.1 User login gets authenticated through firebase		
Pre-Condition: User has already created an account User email and password already in firebase database		
Steps:	Expected Results	Actual Results
1. Input login information	User manages to input email and password	Email and password inputted
2. Click Login Button	Popup should notify user that they were successfully authenticated	Popup notification indicating user was successfully authenticated
Result: Pass		

Test Case: 1.2 User cannot login since they are not registered		
Pre-Condition: User doesn't have an account registered with the app		
Steps:	Expected Results	Actual Results
1. Input login information	User manages to input email and password	Email and password inputted
2. Click Login Button	Popup should notify user login failed since it could not be authenticated	Popup notification indicating user login failed since it could not be authenticated
Result: Pass		

Test Case: 1.3 User can go to the register page		
Pre-Condition: User doesn't have an account registered with the app		
Steps:	Expected Results	Actual Results
1. User taps "register here" text	User will be sent to register activity	User sent to register activity
Result: Pass		

7.2 Test Plan 2: Add Sensor Dialog Fragment

7.2.1 Summary

Story ID: DF-1

Admin Users would like to be able to add a seat sensor to a room they are managing. In this plan, it is desired to test if the dialog fragment works as expected when adding an available sensor to a room.

7.2.2 Test Cases

Test Case: 2.1 Selected room info appears correctly in dialog fragment		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Add Sensor Dialog Fragment.	The Add Sensor Dialog Fragment Opens.	It opened.
2. Check room name and location string at the top of the dialog fragment.	Name and Location of room matches Firebase entry for that roomKey.	Name/Location matches.
Result:		
Pass		

Test Case: 2.2 No sensor message when none are available		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Add Sensor Dialog Fragment.	The Add Sensor Dialog Fragment Opens.	It opened.

2. Check that list of sensors doesn't appear.	The listview with sensor IDs cannot be seen.	It doesn't appear.
3. Check that the “no available sensors” message is displayed.	The textview with the message is visible.	The message appears.
Result:		
Pass		

Test Case: 2.3 Available sensor list updated when sensor is assigned to room		
Pre-Condition:		
<ul style="list-style-type: none"> - There must be a room entry in Firebase with room info already configured (name, location strings). - Test button must be made that opens the Dialog Fragment with a dummy roomKey until the Rooms Activity is completed. - The RoomKey passed to the Dialog Fragment constructor must match one present in the Firebase rooms table until Rooms Activity is completed. - There must be at least 1 sensor entry in Firebase with roomID = 0. 		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Add Sensor Dialog Fragment.	The Add Sensor Dialog Fragment Opens with a list of sensor Keys.	It opened.
2. Click on a sensor ID (remember the ID) in the list to open the add sensor confirmation dialog.	The confirmation dialog opens with the room info (name, location) and “add message” with the right sensor ID.	Confirmation opens with Room info and sensor ID matches.
3. Click on cancel to test cancel button.	App returns to Add Sensor Dialog Fragment with list of available sensor IDs. No changes.	Confirm dialog is closed and listview as well as Firebase data remain unchanged.
4. Repeat step 2 and move to step 5.	<i>See step 2.</i>	Confirm dialog opens with room info and sensor ID that match the one that was clicked.
5. Click on confirm to add sensor to room.	Confirm dialog and Add Sensor Dialog Fragments close, app returns to activity. Entry in Firebase for sensor has its roomID updated to match the room.	App returns to activity, dialogs closed. The sensor roomID in Firebase was updated to the roomKey.
6. Click on the button to re-open Add Sensor Dialog Fragment.	Dialog Fragment re-opens and the ID is no longer in the list.	The sensor ID is no longer in the list

Result:
Pass

Test Case: 2.4 Available sensor list updated when a sensor becomes available		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Add Sensor Dialog Fragment.	The Add Sensor Dialog Fragment Opens with a list of sensor Keys.	It opened.
2. Directly modify the value of roomID for a sensor with an assigned room and change it to 0.	The now-available sensor appears in the list view.	It appears.

Result:

Pass

7.3 Test Plan 3: Profile Activity

7.3.1 Summary

Story ID: Act 2.2

Users would be able to register an account to our app by using their email address and a given password. If the user already has an account they won't be able to register and can go to the login page if wanted.

7.3.2 Test Cases

Test Case: 3.1 User registration		
Pre-Condition: User must not have created an account with their email yet		
Steps:	Expected Results	Actual Results
1. Input registration information	User manages to input email and password	Email and password inputted
2. Tap on the Register Button	Popup should appear notifying user that registration was successful	Popup notification indicating user registration was successful
Result:		
Pass		

Test Case: 3.2 User already has a registered account		
Pre-Condition: User already created an account with their email		
Steps:	Expected Results	Actual Results
1. Input registration information	User manages to input email and password	Email and password inputted
2. Tap on the Register Button	Popup should appear notifying user that registration was unsuccessful	Popup notification indicating user registration was unsuccessful
Result:		
Pass		

Test Case: 3.3 User needs to login

Pre-Condition: User already created an account with their email		
Steps:	Expected Results	Actual Results
1. Tap the “Login Now” text	User will be sent to Login Activity	User sent to login activity
Result: Pass		

7.4 Test Plan 4: Remove Sensor Dialog Fragment

7.4.1 Summary

Story ID: DF-2

Admin Users would like to be able to remove a seat sensor that is already assigned to a room they are managing. This may be to put it in another room, or the seat itself is removed from the room. In this plan, it is desired to test if the dialog fragment works as expected when removing a sensor from a room.

7.4.2 Test Cases

Test Case: 4.1 Selected room info appears correctly in dialog fragment		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Remove Sensor Dialog Fragment.	The Remove Sensor Dialog Fragment Opens.	It opened.
2. Check room name and location string in the dialog fragment.	Name and Location of room matches Firebase entry for that roomKey.	Name/Location matches.
Result:		
Pass		

Test Case: 4.2 No sensor message when the room has no assigned sensors		
Pre-Condition:		
Steps:	Expected Results	Actual Results

1. Click on the button to open the Remove Sensor Dialog Fragment.	The Remove Sensor Dialog Fragment Opens.	It opened.
2. Check that list of sensors doesn't appear.	The listview with sensor IDs cannot be seen.	It doesn't appear.
3. Check that the “no sensors in room” message is displayed.	The textview with the message is visible.	The message appears.
Result:		
Pass		

Test Case: 4.3 Sensors in room list updated when sensor removed from room		
Pre-Condition:		
<ul style="list-style-type: none"> - There must be a room entry in Firebase with room info already configured (name, location strings). - Test button must be made that opens the Dialog Fragment with a dummy roomKey until the Rooms Activity is completed. - The RoomKey passed to the Dialog Fragment constructor must match one present in the Firebase rooms table until Rooms Activity is completed. - There must be at least 1 sensor entry in Firebase with roomID = roomKey that is being tested. 		
Steps:	Expected Results	Actual Results
1. Click on the button to open the Remove Sensor Dialog Fragment.	The Remove Sensor Dialog Fragment Opens with a list of sensor Keys.	It opened.
2. Click on a sensor ID (remember the ID) in the list to open remove sensor confirmation dialog.	The confirmation dialog opens with the room info (name, location) and “remove message” with the right sensor ID.	Confirmation opens with Room info and sensor ID matches.
3. Click on cancel to test cancel button.	App returns to Remove Sensor Dialog Fragment with list of available sensor IDs. No changes.	Confirm dialog is closed and listview as well as Firebase data remain unchanged.
4. Repeat step 2 and move to step 5.	<i>See step 2.</i>	Confirm dialog opens with room info and sensor ID that match the one that was clicked.
5. Click on confirm to remove sensor from room.	Confirm dialog and Remove Sensor Dialog Fragments close, app returns to activity. Entry in Firebase for sensor has its roomID updated to 0 (unassigned).	App returns to activity, dialogs closed. The sensor roomID in Firebase was updated to 0.

6. Click on the button to re-open Remove Sensor Dialog Fragment.	Dialog Fragment re-opens and the ID is no longer in the list.	The sensor ID is no longer in the list.
Result: Pass		

7.5 Test Plan 5: Seating Activity

7.5.1 Summary

Story ID: ACT-1

Users should be able to see a list of rooms that we added and should be able to click on them. To see the list of rooms we need to go to the Firebase server and get the rooms from there, their locations and Seat numbers.

7.5.2 Test Cases

Test Case: 5.1		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. To Print the data from Arduino terminal into firebase	Seat number should decrease when weight is added	Seat number did decrease when weight was added
2. Check which rooms were available on firebase	Check if sensor was available in a room	Available rooms were shown on firebase
3. List rooms on recycler view	The list items should be clickable. And display their results	The List is displaying only dummy profiles for now.
Result:		
Pass		

Sprint 2

7.6 Test Plan 6: App Layout

7.6.1 Summary

Story ID: AL-1

Make the app user-friendly and enjoyable to use by updating the theme including the font family used and colors across the app

7.6.2 Test Cases

Test Case: 6.1 App Layout		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Generate a color palette	A color palette consist of 5 colors is generated	Consistent feel of the app
2. Decide on a font family	Font family selected: Montserrat_light	Consistent feel of the app
3. Update style colors.xml and themes.xml files	The new color palette to be added and applied to the app theme	Fresh new design
Result:		
Pass		

7.7 Test Plan 7: Admin Rooms

7.7.1 Summary

Story ID: PA-3

The administrative users would like to be able to add a room or delete a room from the set of rooms they are managing. In this plan, it is desired to test if the Admin Rooms Activity works as expected for adding or deleting a room.

7.7.2 Test Cases

Test Case: 7.1 Adding a Room		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on toolbar options and click “Add Room”	The Add Room Dialog Fragment opens.	Dialog Fragment opened
2. Enter room Name and Location info for a new room and click Add Room button	A confirmation dialog opens up with the new room info.	Confirm box opened and info matches the input
3. Click on the confirm button.	Dialogs close and the new room appears in the list of admin’s rooms.	Dialogs close and new room appears in list.
Result:		
Pass		

Test Case: 7.2 Removing Room(s)		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Click on toolbar options and click “Remove Room”	The interface is swapped to the Delete mode.	Enters delete mode
2. Click the checkbox beside a room(s) to be deleted.	The checkbox tick appears for the selected room.	The room checkbox is active.

3. Click on the toolbar DELETE button.	A confirmation delete dialog box opens with the info of checked room to be deleted.	The delete confirm opens with the room info as expected.
4. Click on the delete confirm button	The room(s) is deleted and can no longer be seen in the list of admin's rooms. Dialogs close.	All open dialogs closed, room gone, disappears from list.
Result: Pass		

7.8 Test Plan 8: Hardware

7.8.1 Summary

Story ID: HW-1

Hardware needs to directly communicate with the firebase while the sensor is idle and when it is in use.

7.8.2 Test Cases

Test Case: 8.1 Microcontroller provides seat status to user and admin		
Pre-Condition:		
	<ul style="list-style-type: none"> • Microcontroller must be connected to a Wi-Fi signal. • Microcontroller needs to have a minimum of 3.85 volts of input voltage for the Wi-Fi signal to activate 	
Steps:	Expected Results	Actual Results
1. LED will light up to signal user seat is available when powered	Firebase should receive data within a few seconds and continuously output data onto app and the controller's LED should immediately light up	Occasionally, firebase will not indicate on the LED that the seat is taken unless we manually alter the seat's status on firebase for it to work as intended
2. Device must be provided with the minimum amount of power to reduce energy consumption	When provided exactly enough voltage device should work as intended without any issues	Works as intended when given enough power but if the power supply is not steadily supplying voltage the controller will halt
Result:	Pass	

Test Case: 8.2 Microcontroller provides Wi-Fi status to admin		
Pre-Condition:		
	<ul style="list-style-type: none"> • Microcontroller must be placed at an adequate distance from the modem. • Microcontroller should be able to withstand interference from other devices 	
Steps:	Expected Results	Actual Results
1. When device is placed at a reasonable distance	Reading from sensor should be accurate and consistent	Latency and accuracy from readings are received well unless the device is obstructed by multiple walls
2. When introducing a high amount of electrical interference from a powerful computer	It should be able to function normally without any changes in the sensor's data	Only when there is a large signal in direct proximity of the device

		is when the sensor's data becomes inaccurate
Result: Pass		

Test Case: 8.3 Physical hardware's robustness		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. When user sits on seat	Box's frame should not collapse	When user sits on seat, the box does not show any signs of stress or damage
2. When an individual over 50 kg sits on seat	Sensor should not get damaged	When user sits on seat, the pressure point does not apply the entire weight of the person onto the sensor hence there is no threat of damage
3. When sitting on the sensor	The pressure point should not shift from its origin	Pressure point is kept in place with the help of aluminum tape
4. When placing objects onto the seat	The weight of that object should not cause unavailability to users or admins on the app	The weight of anything below 10 kg is almost negligible hence there is no worry for misuse with product
Result: Pass		

7.9 Test Plan 9: Connection

7.9.1 Summary

Story ID: CON-1

Test Plan of a task that involved Sensor Connection Status Which Detected when a sensor is offline or needed attention

7.9.2 Test Cases

Test Case: 9.1 When a sensor is offline, the system should display a message indicating that the sensor is offline.		
Pre-Condition: - Sensor is Online then pre-emptively shut off. - Sensor is assigned to a room		
Steps:	Expected Results	Actual Results
1. Turn off the sensor while it's assigned to a room.	The sensor data should be set to 0	The sensor Data Freezes at the value it was last updated to
2. Check the status of the sensor in the system	The Status of the Sensor should be set to "Offline"	The Sensor Status stays as its last updated value
Result: Fail		
Test Case: 9.2 The sensor should display the connection strength of the WiFi signal.		
Pre-Condition: - Sensor is Online - Sensor is assigned to a room - Sensor is connected to a WiFi network		
Steps:	Expected Results	Actual Results
1. Check the status of the sensor in the system	The arduino should send its sensor data any issue	The WiFi Signal displays its values as it normally does.
2. Check the WiFi Strength of the sensor in various locations	The arduino should send its WiFi Strength as normal.	The WiFi Signal strength is represented in -dBm format (0 to -100).
3. Examine the seat status when the WiFi signal is weak	The arduino should have send its data accurately from far locations	The weak wifi signal transmits its seat sensor values but a longer delay.
Result: Pass		

Sprint 3

7.10 Test Plan 10: Network Configuration

7.10.1 Summary

Story ID: CON-2.3

Admin users want to be able to configure the Wi-Fi network they connect the iSeat sensor. This could be when they initially setup the sensor in a room, if it runs out of batteries and needs to be reconnected, or if they decided to move the sensor to another room that has a different Wi-Fi network. In this plan, we want to test to make sure the sensor Wi-Fi network configuration works and then the device behaves as expected.

7.10.2 Test Cases

Test Case: 10.1 Connect Arduino To Network		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Connect to Arduino Access Point “iSeat” using a Smartphone or a wireless laptop	The computer/phone is connected to the Arduino iSeat network	The phone connects to the iSeat network
2. Open a web browser and open website: 192.168.4.1	The Arduino serves a webpage with input boxes for network SSID and Password	The webpad with network SSID and Password input boxes appear
3. Enter the SSID and password for the Network to connect to and press enter button	The webpage changes to “Attempting to connect to <SSID> with <Password>”, the AP closes, and after a few seconds the device is connected to the given Network	The webpage changes, and a few seconds later the Arduino connects to the network and starts normal iSeat operation
Result: PASS		

7.11 Test Plan 11: Admin/User

7.11.1 Summary

Story ID: ACT-1.7

Setup the activity pages so that admin/user are directed to their designated page

7.11.2 Test Cases

Test Case: 11.1 When an admin or user logs in they will be assigned a role and be directed to their designated page		
Pre-Condition:		
<ul style="list-style-type: none"> When an admin or user signs up, they are assigned unique role 		
Steps:	Expected Results	Actual Results
1. When logged in as admin	They should have access to both admin and user features	Admin has access to all features
2. When logged in as user	They should have access to only user features	User has only access to user features
Result:		
PASS		

7.12 Test Plan 12: Login Indicator

7.12.1 Summary

Story ID: ACT-1.8

Setup a toolbar that has a user icon

7.12.2 Test Cases

Test Case: 12.1 Depending on the status of the person logging in (Admin/User) they will have an icon on the top left of the toolbar that will indicate if they are either an admin or user.		
Pre-Condition:		
<ul style="list-style-type: none"> • Admin/User is logged in 		
Steps:	Expected Results	Actual Results
1. When logging in as an admin	Icon should be an emoticon with a shield	Icon is an emoticon with a shield as intended
2. When logging in as a user	Icon should be an emoticon without a shield	Icon is an emoticon without a shield as intended
Result:		
PASS		

7.13 Test Plan 13: LOG feature for an Admin user

7.13.1 Summary

Story ID: PA-4.3

Admin can have access to a list of users that have created accounts to track how many accounts are existing.

7.13.2 Test Cases

Test Case: 13.1 When admin is checking the		
Pre-Condition:		
Steps:	Expected Results	Actual Results
1. Admin user navigates to the admin profile page	The admin should be able to have access to a different page from the users.	The admin has a selection of cards to choose from which includes the "User List" card
2. Admin user clicks on the "User List" card	A log of users who have created accounts on the system is displayed, including their username, the date/time of their account creation, their role and their email.	All of these are shown to be present in the list of created users.
Result: PASS		

VIII. ETHICAL DIMENSIONS

8.1 Individual and Society

In considering the ethical dimensions of the Seat-Sensor product, it is important to consider the potential impacts on both the individual and society. From the perspective of the individual, the Seat-Sensor app has the potential to limit the activities of students who may occupy seats with their bags, potentially violating other people's right to use available seating. Additionally, the real-time location data collected by the app could potentially violate students' rights to privacy if accessed by unauthorized individuals.

From a societal perspective, the Seat-Sensor app could have legal implications, such as the need to comply with data privacy laws and regulations. It is also important to consider the potential public policy implications of the widespread use of the Seat-Sensor app, such as the potential for increased surveillance in schools. As a professional engineer, there is a responsibility to consider product liability and ensure that the Seat-Sensor app is safe and does not pose any risks to students or staff.

8.2 Stakeholders

In terms of stakeholders, the Seat-Sensor product affects multiple groups, including students, teachers, and administrators. The interests of these stakeholders may not always align, such as when students may be more focused on convenience and teachers on maintaining a safe learning environment. The Seat-Sensor app could potentially impact situations where stakeholders' interests are not aligned, such as by providing real-time data on student locations that could be used to manage and prevent conflicts.

Overall, it is important to consider the ethical dimensions of the Seat-Sensor product, including individual and societal impacts, legal and policy implications, and the interests of different stakeholders. By taking these considerations into account, the Seat-Sensor app can be designed and implemented in a way that promotes the safety and well-being of students and staff.

8.3 Potential Ethical Considerations

Product misuse: Some students may place their bags on seats to make room for others, potentially causing the Seat-Sensor to misidentify the number of occupied seats.

Solution(s): Increased the Seat-Sensor's threshold for distinguishing between an average student's body weight and an object weighing less than that value.

Data security: The Seat-Sensor app implies the real-time location of students within the school, which could be potentially dangerous if accessed by unauthorized individuals, such as a school shooter. In the case of the school shooter example, it could provide them information on the most crowded areas.

Solution(s): To prevent this, the app could request permission from the user to access the phone's Wi-Fi module and ensure that the collected data is kept private. Furthermore, a strict policy could be put in place to ensure that the data collected is only accessed by authorized individuals, such as school administrators and security personnel.

User data: It's important to inform users that the data collected by the Seat-Sensor app is private and will only be used for the intended purpose of identifying occupied seats.

Solution(s): Users should be notified that the data will be deleted once their school years are over, or when the user decides to opt-out of using the app.

Theft/Vandalism: There is a potential risk of students stealing or breaking the Seat-Sensor setup, which could disrupt the functionality of the system.

Solution(s): Schools could take necessary steps to secure the Seat-Sensor equipment, such as by installing security cameras in areas where the sensors are located, or by implementing a system for tracking and reporting missing or damaged equipment. Schools could also inform students of the consequences for theft or vandalism, such as disciplinary action.

Arguments and Conflicts: Students heading to the same room or seat may spark arguments or conflicts among themselves. It's important to monitor and address these situations to maintain a positive and safe learning environment. Also, some students might find out their seat has been taken while they walked away from their seat momentarily.

Solution(s): The Seat-Sensor app could potentially provide data on student locations that could be used to prevent or manage conflicts, such as by directing students to a different room or seat that meets his preferences. And to combat the issue on the seats being taken while a student is taking a break from sitting, we can add a delayed amount of time for the seat to be recognized as available again.

IX. COMPUTER SIMULATION

Initial Computer Simulation Plan

As discussed in Milestone 2, our first idea for a computer simulation consisted of making 3D models and testing the seat coverage of the sensor circuit. This was to figure out if we should use the Film sensor or the Load Cell sensor, and if 1 sensor would be enough, or two sensors, 3 sensors in a triangle shape perhaps, or four in a square, etc. We proceeded in this direction, and created models on SolidWorks for a seat as well as several of the above-mentioned potential sensor configurations. After working with the Models for a while, we discovered there wasn't much data to be had from this modelling, it seems that any number of sensors would work with the product, either with Film Sensor or Load Cell, and so the results of the initial simulation were *inconclusive*. This led us to revise our simulation idea and perform another one as detailed below.

Revised Computer Simulation Plan

The revised simulation plan was to help us decide if we should be using the Film sensor or the Load Cell sensor with amplifier in our final product. To make this choice, we used the Proteus 8 Design Suite software to create simulations of both Arduino circuits. As a lucky break, the Load Cell and HX711 amplifier test circuit were included as default Arduino examples in Proteus, which allowed us to quickly get a simulation running for the Load Cell circuit. The Film sensor was more tricky, as that part could not be found either in the default Proteus part library or in online libraries. We ended up creating our own part for the Film sensor, as it is basically just a variable resistance, and the formula governing its behavior can be found on a resistance-vs-weight graph from its datasheet. All that remained was to write a sample Arduino sketch that could test the sensors, upload it to the simulated Arduino, and start examining the type of output we would expect to get from each of the sensor to make the choice of final sensor to use for the product.

Simulation Results Interpretation

The results obtained from the simulations favored the Load Cell as the sensor of choice. This is because the Load Cell was able to detect the simulated weight reliably up to the desired ~50 kg range, while the Film sensor functioned basically like an ON/OFF switch. As soon as the Film sensor had a little bit of weight applied to it, its resistance changed dramatically, going from Open-circuit (no weight) to Short-circuit (max weight) almost immediately. It could not offer the precision necessary to sense the weight applied to it, and maxed out around 20 kg. Clearly this would not be enough for us to be able to test the applied weight to the sensor, to determine if it is a person sitting on the chair or simply a bag/other stuff for our application. The results were further confirmed during testing: using a circuit with the Film Sensor provided weight data that jumped around and never gave the right weight value; while using a circuit with the Load Cell gave a steady weight value we could use for the product. Therefore, the simulation concluded with us deciding that we shall use the Load Cell sensor in the iSeat product, and following that the prototype seat sensor was developed.

X. PRODUCT PITCH

See the media submission with this report for the pitch video of the iSeat product.

Script:

1. “Hello, and Welcome to the iSeat”

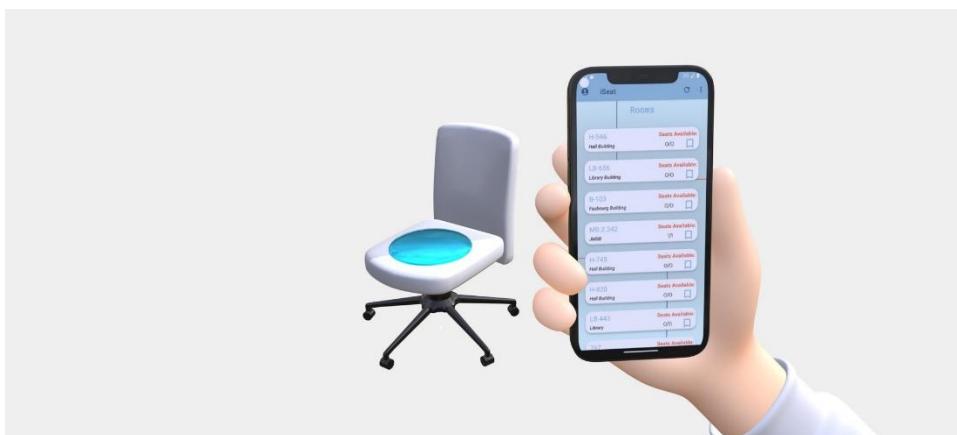
<background is ppt intro slide with product name/logo, slogan, team members and Concordia course>
(max 5 seconds so ppl have time to read info slide)



2. “iSeat is the newest in seat tracking technology, allowing students in universities to have real-time updates on seat availability in their favorite study rooms.”

“This is accomplished by using IoT enabled seat controllers that interface with a remote server, whose data is then conveniently listed on an Android smartphone using the companion app.”

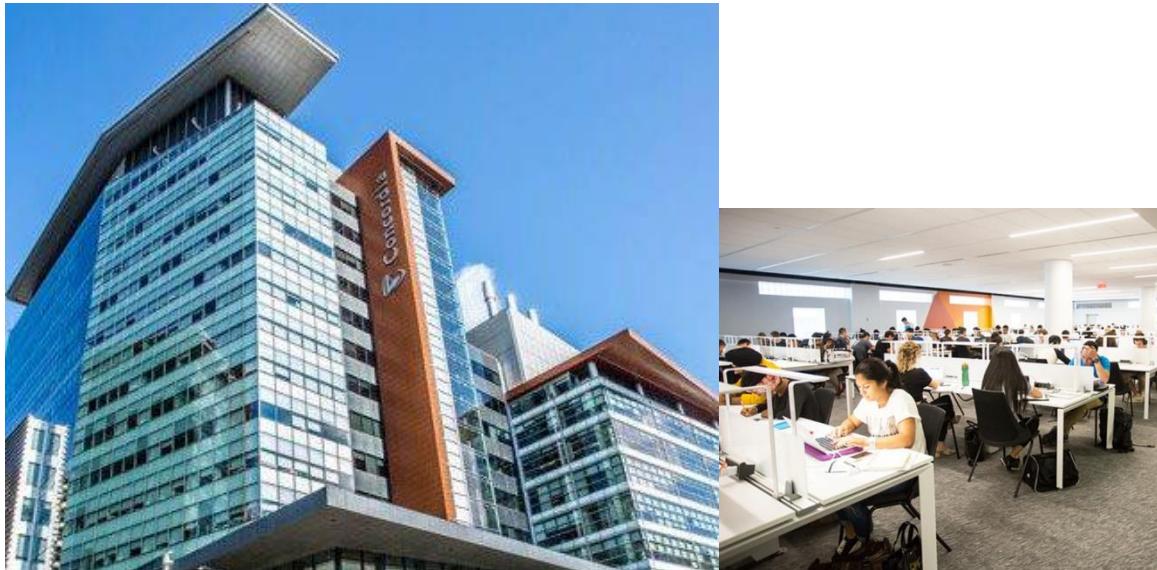
<background pic: maybe zoom in a little, maybe add animation like ARROW POINTING when seat sensor is mentioned, and then when Smartphone app is mentioned>



3. “Primarily targeted at institutions of higher education like Universities and CEGEPs, the iSeat system is designed to reduce their student body’s frustration as finding a free seat in a room to study can be a challenge.”

“Instead of wasting time walking around the multiple buildings on campus, students can take a look at their favorite rooms and see if there is an available seat left for them, and then immediately make their way there.”

<Maybe a pic of Concordia building outside with name visible, with an inset of library study room with all the seats showing, something like below>



4. “iSeat is looking for investors to help get the product off the ground and scale up production to handle the needs of large clients like Concordia or other universities”

“With development costs of \$21,600.00, a per-unit cost of \$5.90, and a unit price of \$15.00, the projected amount of units sold will have to be 2,373.62 units before generating any profit. Assuming a large University like Concordia has around 5000 seats, the project profits for such a contract would be around \$24,000.00.

<put the calculations in slide with nice background>

Development costs: \$21,600.00

Per unit cost (for us to make): \$5.90

Per unit price (for client to buy): \$15 (bulk prices available)

Sales needed for profitability: $21,600 / (15 - 5.90) = 2,373.62$ units

Profits for Uni with 5000 seats: $(5000 - 2,373) * \$15 = \$23,905.7$ profits

XI. FINAL ORAL PRESENTATION SLIDES

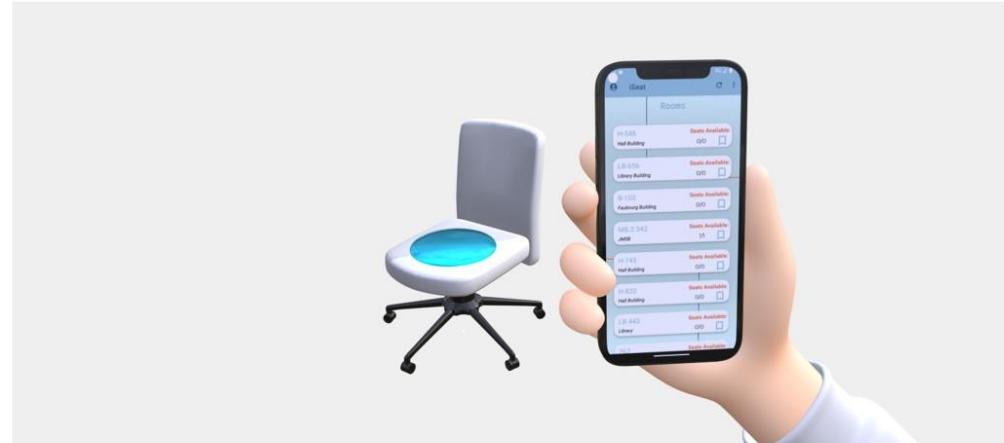


iSeat
iSeat, you seat, we all find a seat

COEN/ELEC 390 Project
Developed by **Team D:**

Adnan Saab	Samson Kaller
Farah Salhany	Shahin Khalkhali
Marwan Al-Ghaziri	Shayan Khalkhali

Concordia University



A Seat Tracking App

- Uses sensor to detect seat availability
- Provides a list of rooms to direct the user to a nearby seat.

Target Market

Higher Education Institutions & General Population



Stakeholders



The Customer

The Administration at Concordia University
IT support



The Users

General population that live near the Concordia
University Library
Concordia University students



The Customer

- Will have admin privileges which are the following:
 - Access to room availability
 - Access to their avatar and personal information
 - Access to room management
 - Access to the user list
-



The User

- Will have user privileges which are the following:
 - Access to room availability
 - Access to their avatar and personal information
 - Access to favorite rooms
-

Product Value

User Benefits

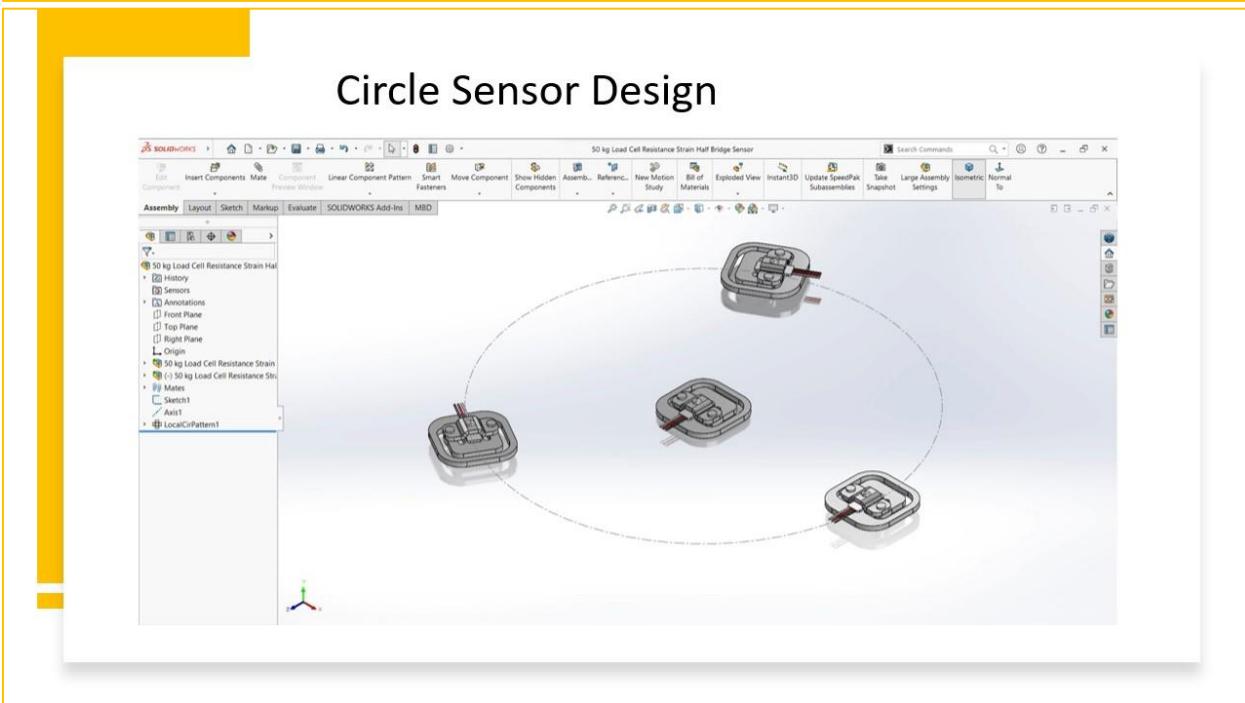
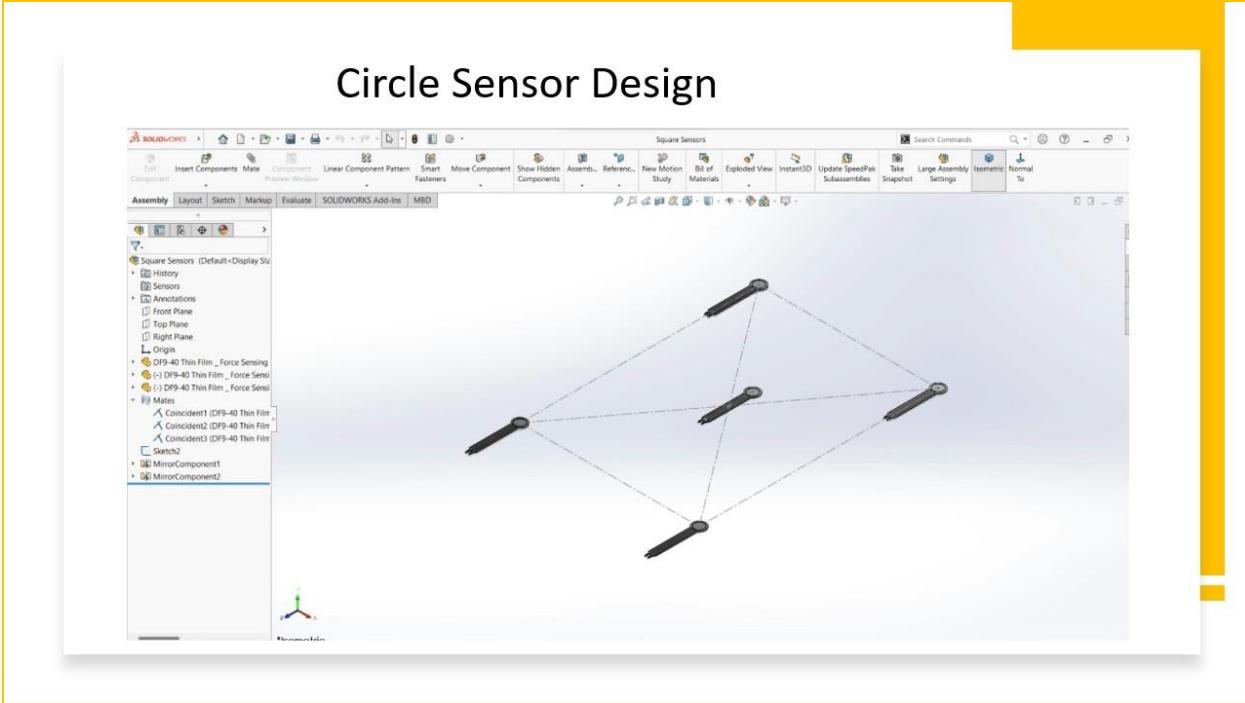
- Real Time Update
- Facilitate finding seats for students
- Start studying faster
- Save time

Customer Benefits

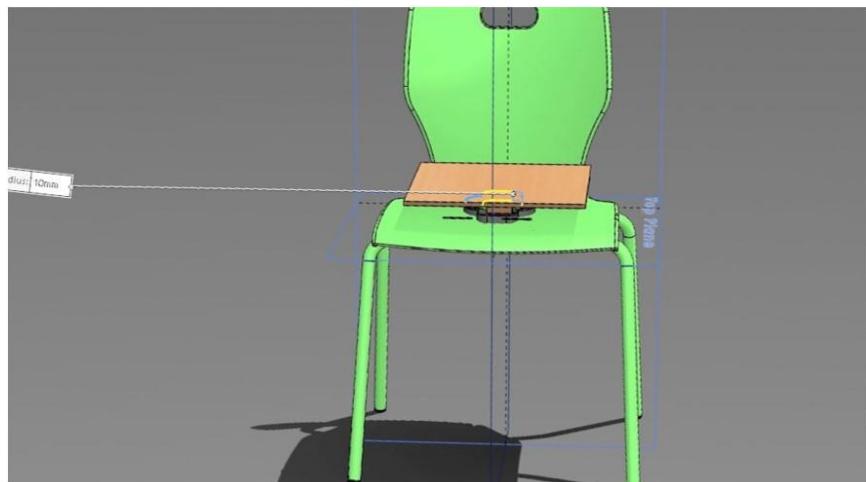
- Reduces student's frustration
- Easy to implement
- Add multiple rooms
- Add/delete seats

Will be more useful and accessible than Concordia's current study room reservation app

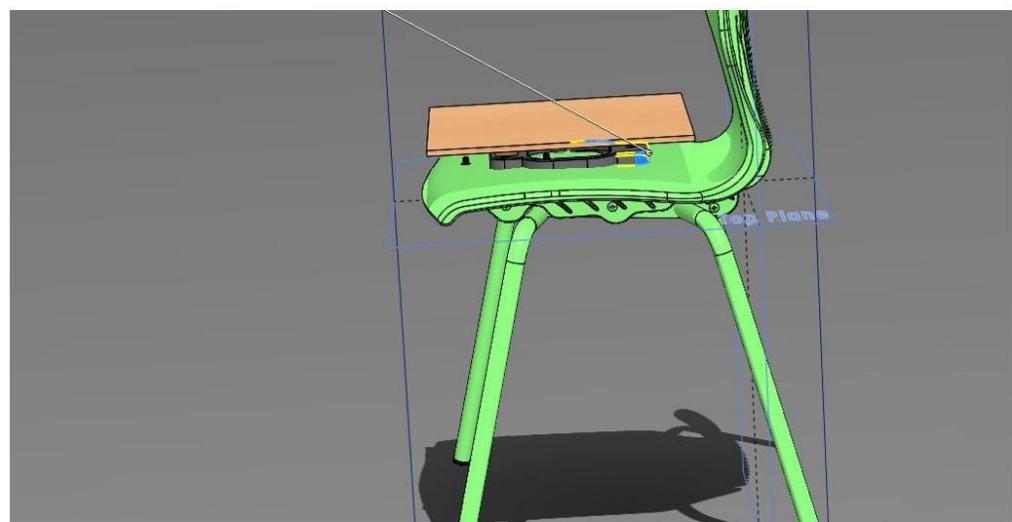
Early Product Designs



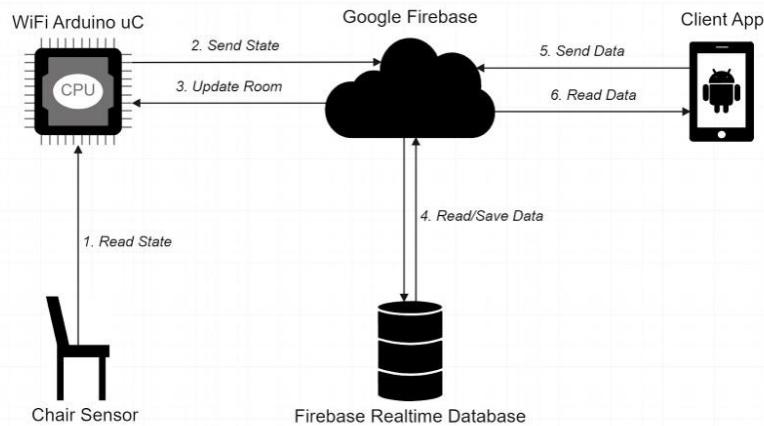
Solidworks View of Square Design



Solidworks View of Square Design

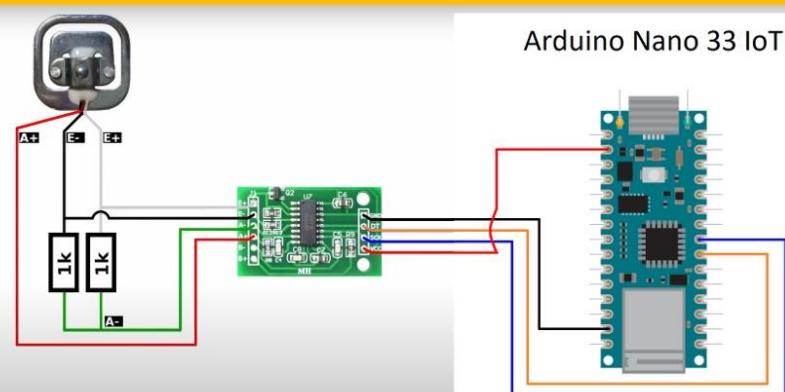


Solidworks View of Circle Design



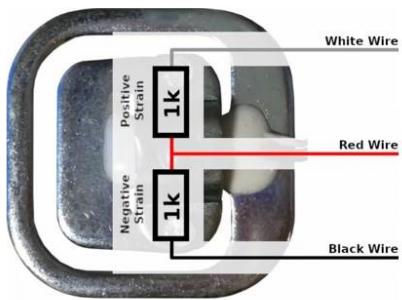
iSeat System
Architecture

iSeat Circuit Schematic



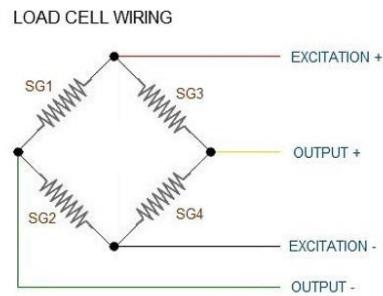
The Wheatstone Bridge

Half Bridge Wheatstone



circuitjournal.com

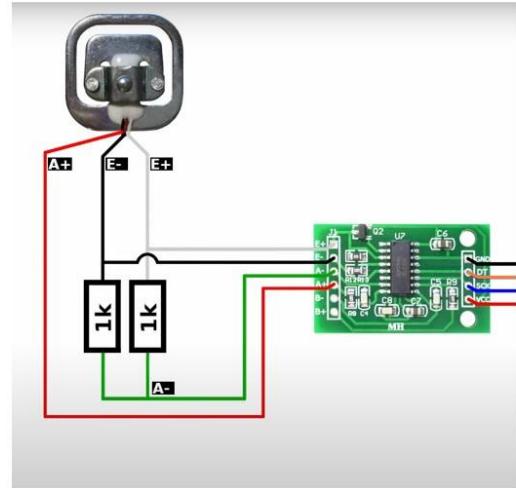
Full Bridge Wheatstone



iSeat Circuit Schematic

- Sensor → Amplifier

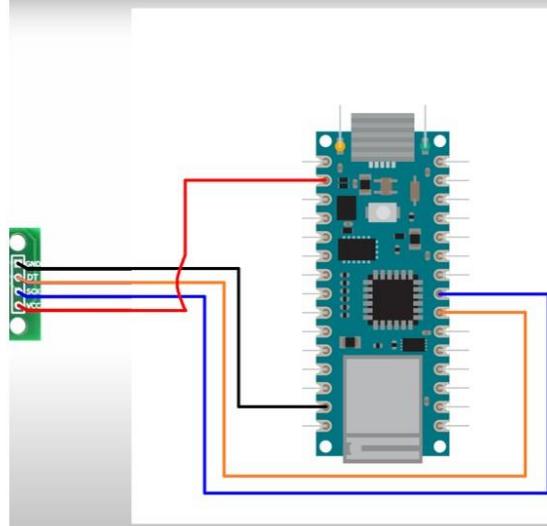
- Load sensor measures force electronically when stress applied at fulcrum via “sense voltage”
- Add two $1\text{k}\Omega$ resistors to construct the full bridge wheatstone
- HX711 amplifies small analog signal and converts it to a digital signal



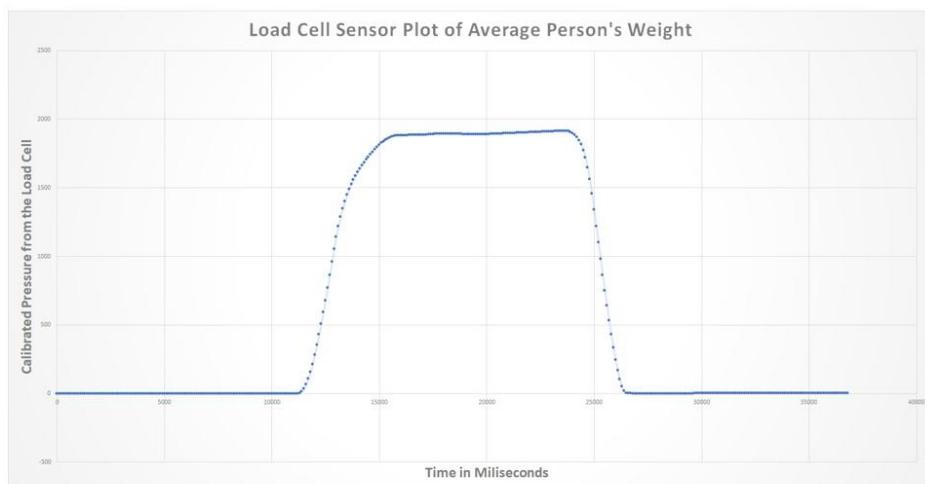
iSeat Circuit Schematic

- Amplifier → Microcontroller

- The microcontroller is powered by an external vin
- The amplifier is powered by the microcontroller
- The digital signal is received into the digital pins and is read

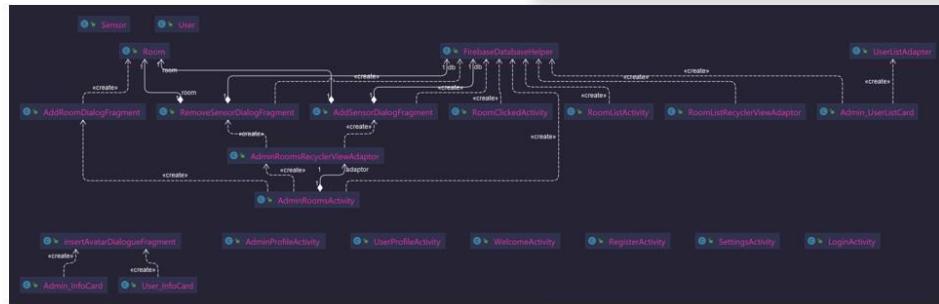
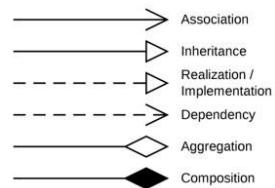


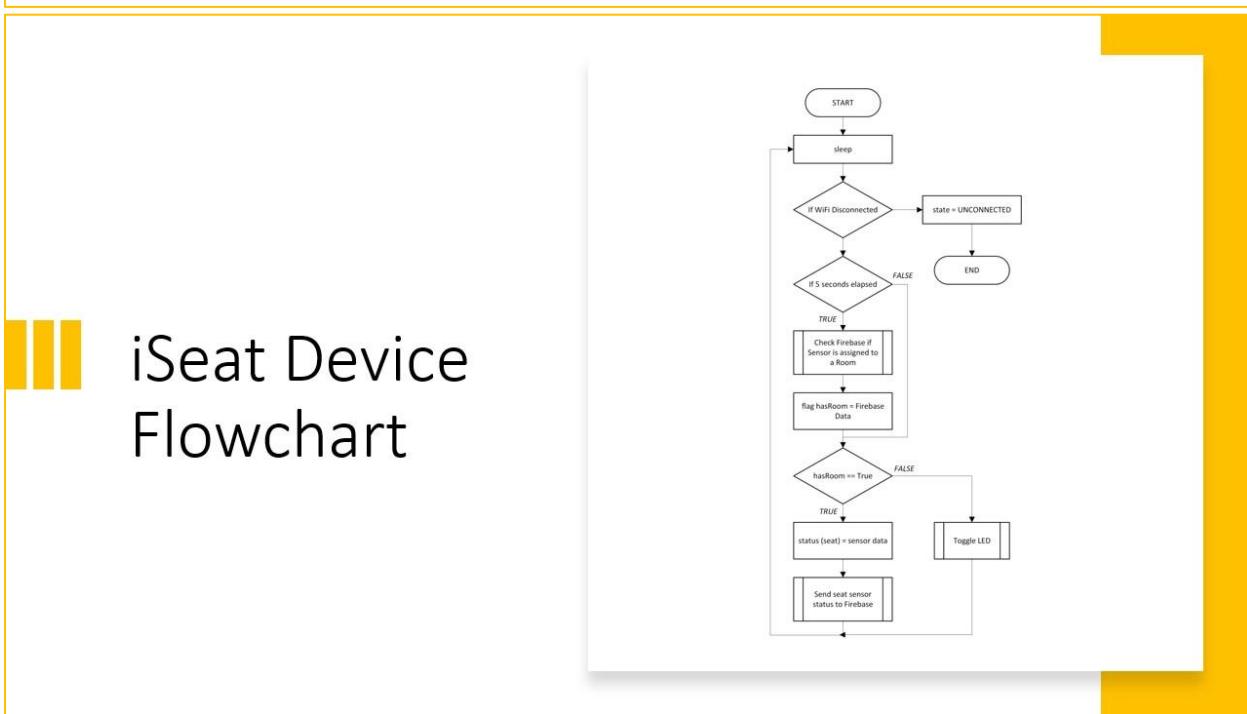
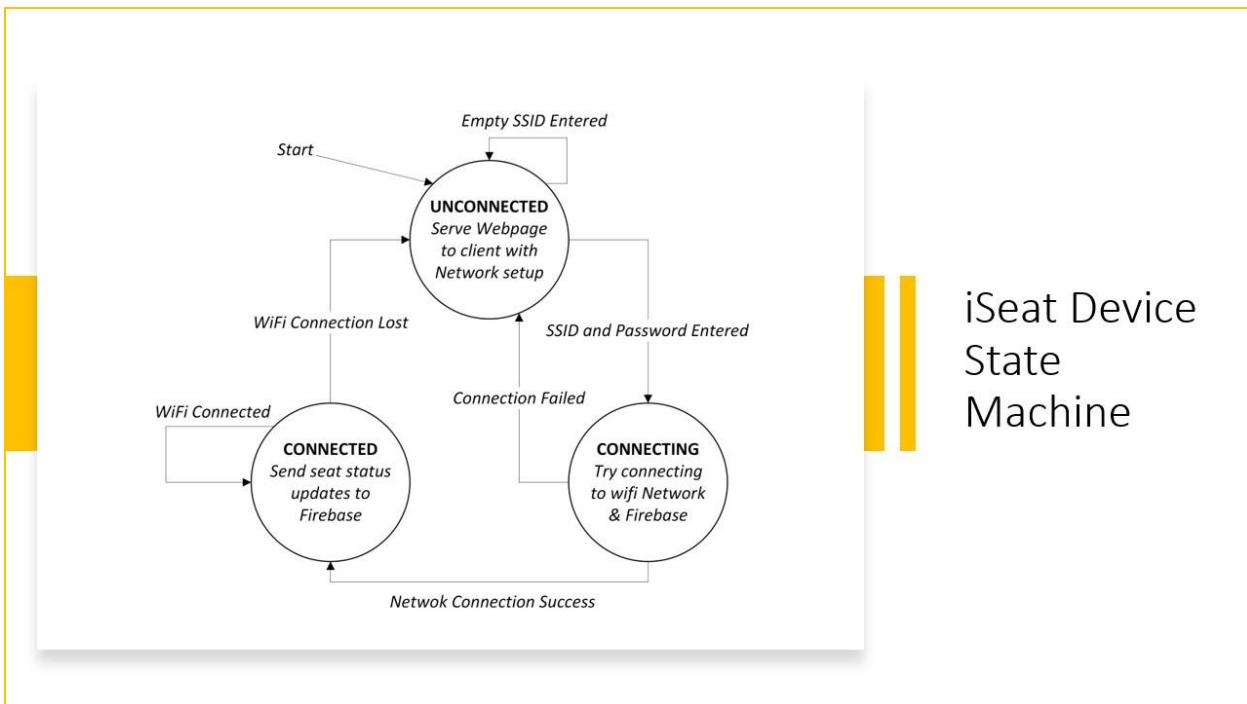
Load Cell Graph



Software Architecture

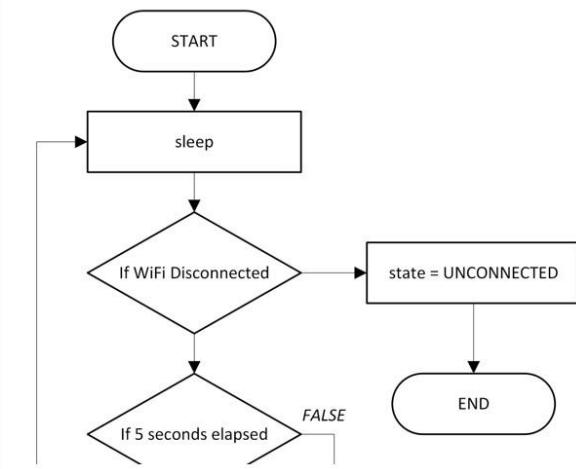
A hierarchical UML chart of our project.



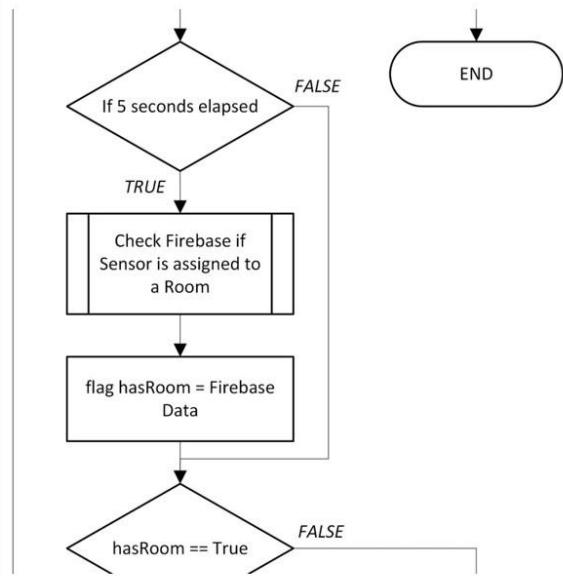




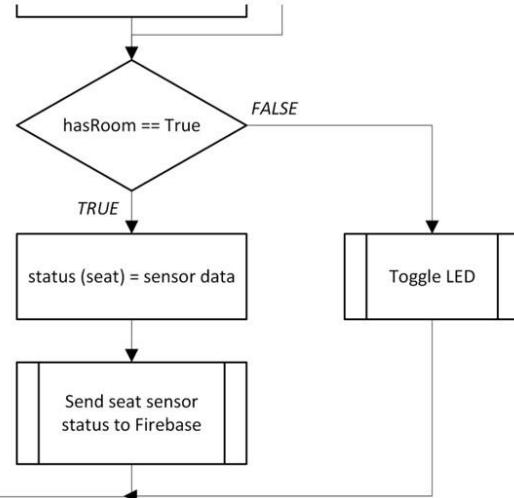
Flowchart




Flowchart



Flowchart



iSeat Android App

iSeat is a fast and fun way to find available seats in your favorite space.

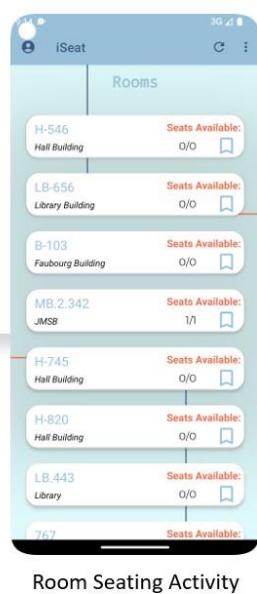
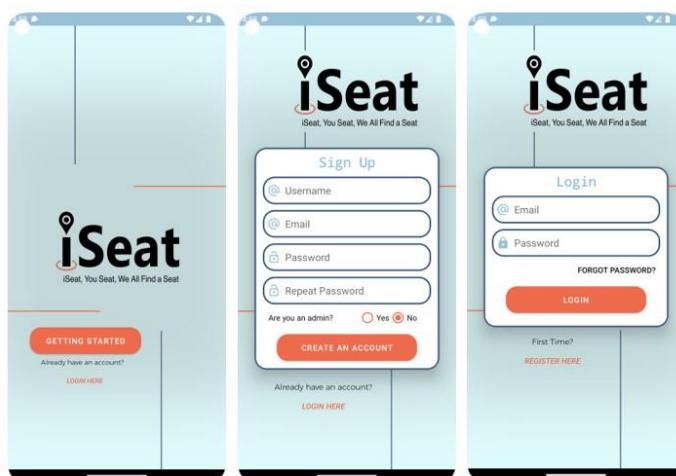
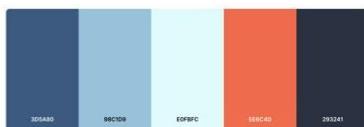
Features:

- Simple user-friendly layout and intuitive to use
- Account creation and deletion at any time
- Dark mode option
- About buildings and their location.



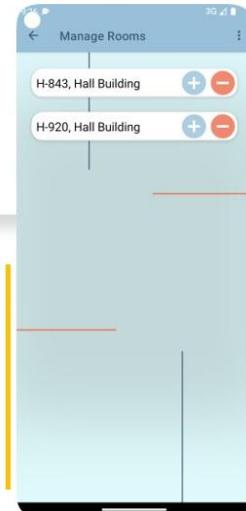
Registration and Login

Color palette

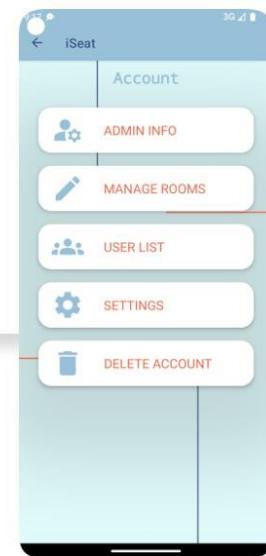


App Activities

Admin Room Management Activity



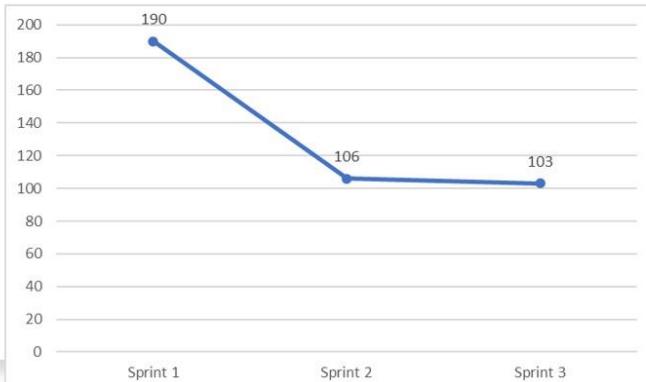
Room Seating Activity



Other Activities

Note*
Sprint 1 Story
points were
overly estimated.

Project Velocity



Average Velocity: 133 Story Points / Sprint



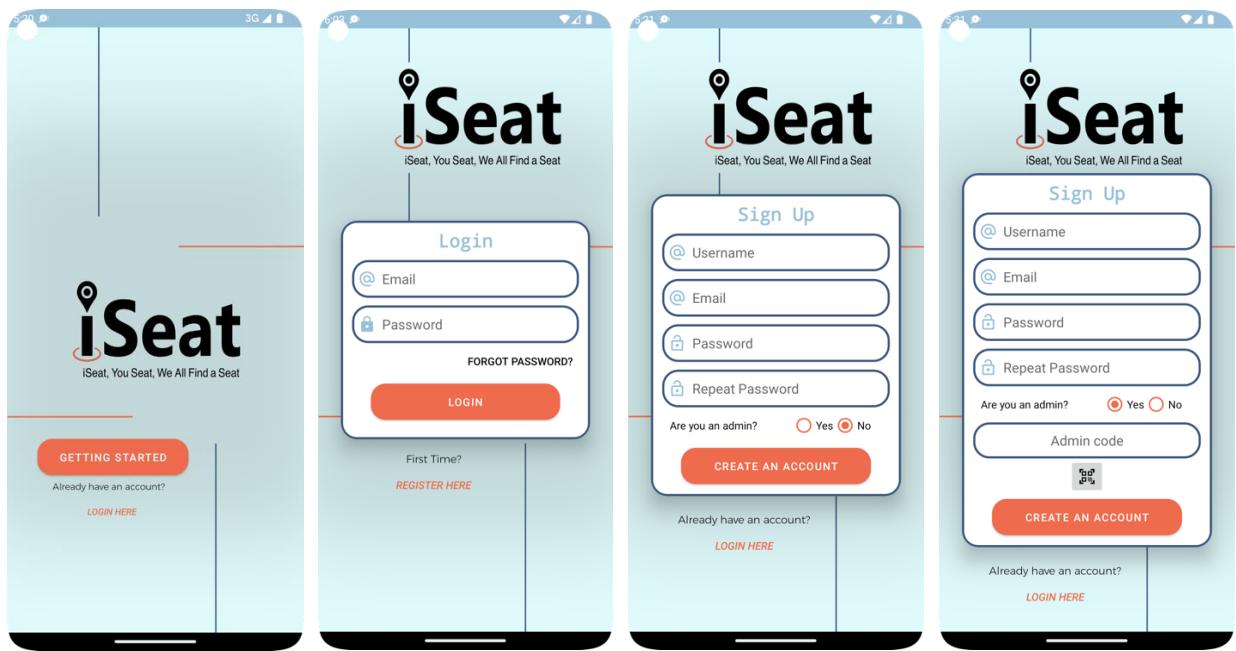
Thank you for listening!

iSeat, you seat,
we all find a seat

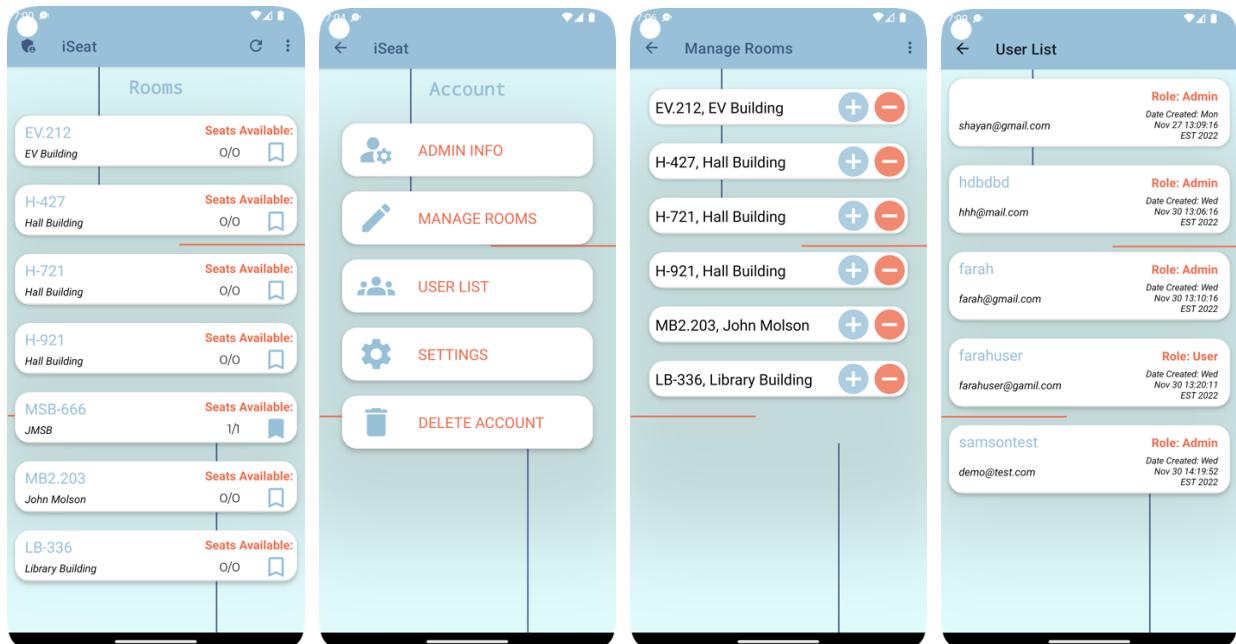


XII. PRODUCT APP

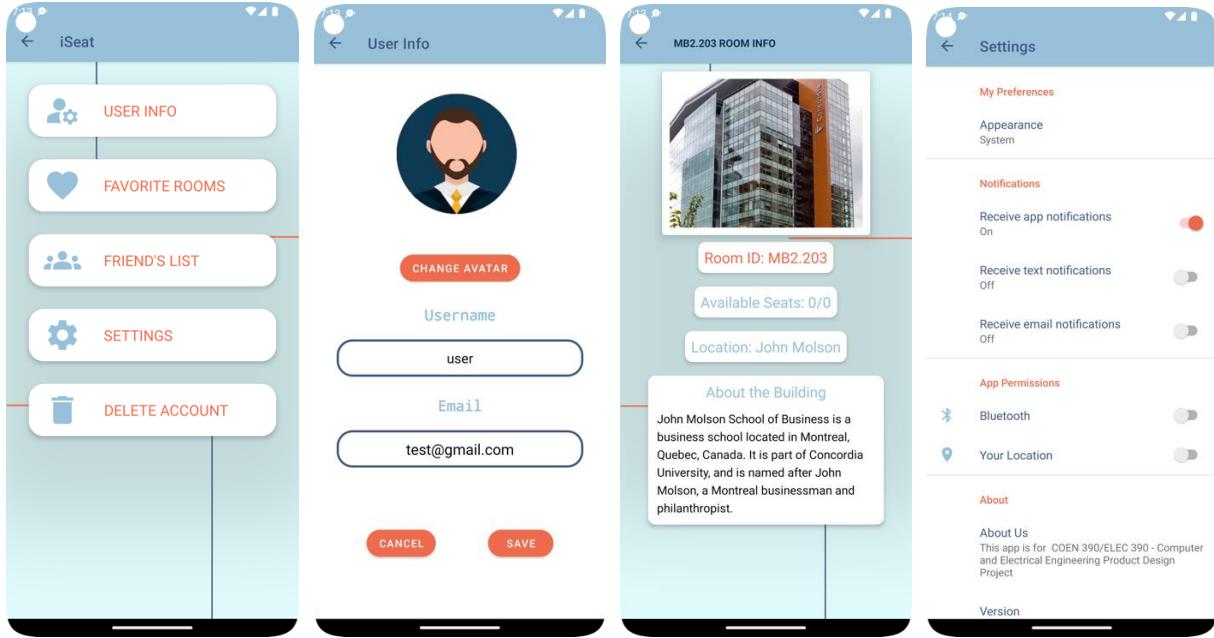
Registration & Login in



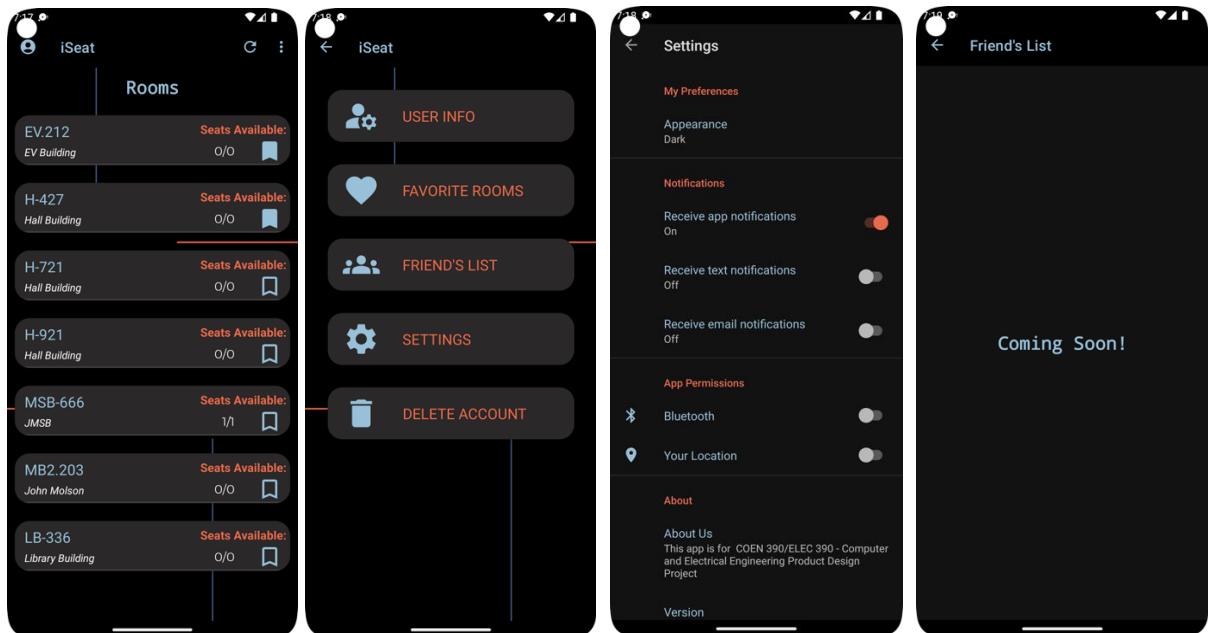
App Activities for Admins (Rooms List, Account Settings, Managing Rooms, and Users)



App Activities for Users (Rooms List, Account Settings, Managing Rooms, and Users)



App Activities in Dark Mode and future features



XIII. FINAL TEAM BLOG

As a team, we have agreed amongst ourselves that this is an accurate representation of what our team has done for the duration of this course.

Team Members					
First Name	Last Name	Nickname			
Adnan	Saab	AS			
Samson	Kaller	Sam			
Farah	Salhany	FS			
Shahin	Khalkhal	Shah			
Shayan	Khalkhal	Shay			
Marwan	Al-Ghaziri	MA			

Activities Table

Date	Who					Type of Activity	Number of hours spent	Purpose	Output	Hours spent				
	AS	Sam	FS	Shah	Shay					AS	Sam	FS	Shah	Shay
Milestone 1														
24-Sep-22	1	1	1	1	1	meeting	3	Think of opportunities and create table of evaluation	brainstorming, writing opportunity statements and evaluation	3	3	3	3	3
26-Sep-22	1	1	1	1	1	meeting	2.5	Set mission statement	ranking opportunity statements and writing mission statement	2.5	2.5	2.5	2.5	2.5
27-Sep-22		1				Report	1.5	Format output of previous 2 activities into a pdf report	Milestone 1 pdf report	0	1.5	0	0	0
										Total hours	5.5	7	5.5	5.5
										Total team hours	2.5			31.5
Milestone 2														
11-Oct-22	1	1	1	1	1	meeting	1.5	Create draft interview script	Draft interview script	1.5	1.5	1.5	1.5	1.5
11-Oct-22			1			work at home	0.5	Prepared interview template	Help improve quality of meeting	0	0	0	0.5	0
12-Oct-22	1	1	1	1	1	Meeting	0.5	Interview: Alex @ 2h45pm	Important Outcomes of interview	0.5	0.5	0.5	0.5	0.5
13-Oct-22		1				Documentation	1.5	Setting up Github repo and Kanban Board for out project management in agile project		0	0	1.5	0	0
14-Oct-22			1			Documentation	0.5	Added links and comments to the Github repo Backlog		0	0	0	0.5	0
15-Oct-22				1		Documentation	1	Transcribed interviews into text		0	0	0	0	1
15-Oct-22	1	1	1	1	1	Meeting	0.5	Interview: Jane Doe @ 1:45pm	Important Outcomes of interview	0.5	0.5	0.5	0.5	0.5
15-Oct-22		1				Documentation	1.0	Computer Simulation Plan	Computer Simulation Section of report & Discovery work stories in PB	0	1	0	0	0
16-Oct-22	1	1	1	1	1	Documentation	4.0	Finishing up Milestone 2 Report sections: PB and Ethical Dimensions	Milestone 2 sections finished	4	4	4	4	4
16-Oct-22		1				Documentation	0.5	Formatting Milestone 2 parts into final pdf	Final version of Milestone2 PDF.	0	0	0.5	0	0
										Total hours	6.5	7.5	8.5	7.5
										Total team hours	6.5			44
Sprint 1														
17-Oct-22	1	1	1	1	1	Meet	3	Plan sprint 1	Sprint 1 goal. Sprint backlog	3	3	3	3	3
21-Oct-22		1		1		Hardware	1	Pickup sensor	Be able to begin working on project	0	1	0	1	0
24-Oct-22	1	1	1	1	1	Meeting	1			1	1	1	1	1
26-Oct-22	1		1			Hardware	4	Connect sensor to arduino	Improve readings of sensor data	0	4	0	4	0
27-Oct-22		1		1	1	Hardware	5	Send densor data from microcontroller to firebase	Create real time database & create code to send sensor seat state to firebase	0	5	0	5	0
27-Oct-22				1		Seats Page	4	To create rooms list visible to the user in the app	Created list using recyclerView with dummy rooms	0	0	0	0	4
28-Oct-22				1		Seats database	4	Create firebasedatahelper	to allow firebase to read data from the app to firebase	0	0	0	4	0
28-Oct-22					1	Study Room & Sensor Database	4	To setup databases for rooms and sensors	Have SQL-lite databases for both Rooms and sensors	0	0	0	0	4
28-Oct-22				1		Manage Rooms Page	3	Create Room Management Activity	Admin can add study rooms and seats	0	0	0	3	0
30-Oct-22		1				Documentation	3	Design Document	Android Application Wireframes section. System Architecture section. Setup template for test cases section.	0	3	0	0	0
31-Oct-22	1	1	1	1		Meeting	0.5	Work on sprint report		0	0.5	0.5	0.5	0
31-Oct-22	1			1		3D Printing	1	3D print frame for the load cell	Load cell will be kept in place for the prototype and final model	1	0	0	1	0
31-Oct-22					1	Learning SolidWorks	3			0	0	0	0	3
31-Oct-22					1	SolidWork Design	3			0	0	0	0	3
31-Oct-22				1	1	Design Miniature Prototype	1	Create a miniature prototype to present	demonstrate the product's functionality	0	0	0	1	0
1-Nov-22	1	1	1	1	1	Documentation	2			2	2	2	2	2
										Total hours	7	19.5	6.5	25.5
										Total team hours	12			91
Sprint 2														
3-Nov-22		1				Coding	5	Working on Admin Rooms Activity, refactoring other files	Admin Rooms ACtivity	0	5	0	0	0
4-Nov-22		1				Coding	2	Admin Room Activity, working on getting/setting data in firebase, created test buttons on main page, refactor to recyclerview adaptors in app	Admin Room Activity with connection to Firebase	0	2	0	0	0
5-Nov-22					1	SolidWork Design	4			0	0	0	0	4
5-Nov-22		1				Coding	2	Refactoring	converted AddSensor Dialog Fragment to use the FirebaseDatabaseHelper class	0	2	0	0	0
7-Nov-22					1	Researching	2.5	Allow RecyclerView to list rooms sorted by favorites	Learned the approach to take	0	0	0	0	2

8-Nov-22					1	Researching	2	Allow RecyclerView to list rooms sorted by favorits	Learned the approach to take	0 0 0 0 0 2
9-Nov-22		1				Coding	2	Refactoring	Converted RemoveSensorDialogFragment to use FirebaseDatabaseHelper class	0 2 0 0 0 0
10-Nov-22		1				Generate Color Pallete/ elements design and choosing a font	3	get consistent look throughout the app	Choosing a color pallete (5 colours) to use and 2 fonts	0 0 3 0 0 0
11-Nov-22	1	1	1	1	1	Meeting	1			1 1 1 1 1 1
12-Nov-22			1			Design Prototype	3	Design box to contain hardware contents and frame for seat	Created frame using wine box	0 0 0 3 0 0
13-Nov-22			1			Design Prototype	5	Create cushioning for seat, assemble metal plate with pressure point, and add layer of wood	Dampen stress on sensor, give feedback to system, add height to reach pressure point.	0 0 0 5 0 0
13-Nov-22					1	Researching	2.5	Research	Learned the approach to take	0 0 0 0 0 2.5
13-Nov-22					1	Coding	4	Worked on allowing rooms to be listed by favorits	In progress	0 0 0 0 0 4
13-Nov-22			1	1		Allow Arduino and Sensor to function without being plugged in.	2	Making the Arduino and Sensor work without being directly plugged into the PC and only using an external power source.	Figured the Amout of Voltage required to Power a single set of our Product.	0 0 0 2 2 0
13-Nov-22					1	Wifi signal strength used to indicate connection status.	3	To verify if the Sensor is able to properly/accurately transmit it's Data	The Arduino can send its WiFi Strength Status to the FireBase DataBase and indicate if the connection is reliable or not. Also discovered that large sources of noise interference (for example, a computer) have an effect on sensor data.	0 0 0 0 3 0
13-Nov-22		1				Updating the app activities with new style and theme	6	Improve app layout and user experience when using the app	improved Welcome/Sign up/Sign in activities	0 0 6 0 0 0
13-Nov-22	1					Coding	8	Coded AddRoomDialog Fragment and Feature for removing rooms	Admin add room and delete room feature	0 8 0 0 0 0
12-Nov-22	1					Facilitate navigation in app	4	side menu that allows users to easily go through different activities	Side pop up navigation menu	4 0 0 0 0 0
14-Nov-22					1	Coding	4	Worked on allowing rooms to be listed by favorits	In progress	0 0 0 0 0 4
14-Nov-22	1					testing firebase delete profile	5	Sync updates on user profile in firebase server	email and password removed from database	5 0 0 0 0 0
14-Nov-22	1					Implementing profile delete in user settings	5	Give user a choice to delete profile from server	Button to delete user profile	5 0 0 0 0 0
14-Nov-22			1	1		Design Prototype	3	Finishing touches and testing	Glued/taped everything into place and tested the final product	0 0 0 3 3 0
14-Nov-22		1				Adding toolbar + action items	3	being able to have more options once in the Room activity	user have more options that are easily accessible from the toolbar	0 0 3 0 0 0
14-Nov-22	1					Coding	4	Bug fixing, app navigation fixing, prepare app for Sprint Demo	Sprint demo	0 4 0 0 0 0
15-Nov-22			1			Documentation	1	Created template for sprint 2 wrap up	Made documenting easier for team	0 0 0 1 0 0
15-Nov-22				1		Added a Profile Icon on the User Page.	3	To notify the user that he is signed in and to provide an easy way to access his profile page.	The icon appears as intended, but the profile page it directs to is still incomplete and in need of some additional features and enhancements.	0 0 0 0 3 0
15-Nov-22		1				Settings activity	3			0 0 3 0 0 0
15-Nov-22	1	1	1	1	1	Documentation	3	Sprint 2 wrap up/Sprint 3 planning	Sprint 2 document + Sprint 3 planning	3 3 3 3 3 3
16-Nov-22				1	1	Testing	1	Tested prototype again with boards being powered by a battery To see if the Sensor can transmit data without being connected to a computer.	When we used a proper power source, we discovered that our circuit board and code were working properly.	0 0 0 1 1 0
Total hours Total Team hours									18 27 19 19 16 22.5	121.5

Sprint 3	AS	Sam	FS	Shah	Shay	MA					
17-Nov-22	1	1	1	1	1	1	Meeting	1	Discuss sprint 3 goals	Assigned tasks to each member	1 1 1 1 1 1
19-Nov-22		1					Coding uC	5	Creating state machine with Unconnected, Connecting, and Connected states	Prototype Code for Unconnected State, uC works as AP and asks user to input SSID and PASS for network	0 5 0 0 0 0
19-Nov-22			1				Dark Mode setup	4	create a dark mode theme and colors	Theme and colors updated	0 0 4 0 0 0
20-Nov-22			1				Dark mode implementation	2	update all attributed to implement the dark theme	App dark mode implemented	0 0 2 0 0 0
21-Nov-22				1			Card activity layout	5	Create layouts and activities for user icon toolbar	Allow user to get access to their information, favorite rooms, history, friend's list, settings, and delete account	0 0 0 5 0 0
21-Nov-22					1		Building info list	4	Create layout for each room clicked	Clearer description of text view, and a description about the building sensor is located in	0 0 0 0 4 0
22-Nov-22					1		Display Connection Status To Admin	3	The admin will be able to see the connection strength of the sensors that are online	The arduino will send its data to the firebase then it will send it to the app only for admins so they can diagnose any issue with the sensor	0 0 0 0 3 0

22-Nov-22				1		Interrupt bugs	3	Interrupt in the Load Cell routine continuously functions to display the data accurately to the Firebase.	Checks to see how the Firebase database reacts and how the app behaves when a sudden power disconnects occurs midway from the arduino/sensor	0 0 0 0 3 0
23-Nov-22			1			Room Info Management	2	Admin is able to Add information based on the rooms features	Can be features like "has a whiteboard" or "has Projector"	0 0 0 2 0 0
24-Nov-22		1				Setting Activity functionality	3	Make the setting page functional and save the preferences in SharePreferences	App saves user settings when quitting the app and resuming	0 0 3 0 0 0
24-Nov-22				1		Favourite Code	4			0 0 0 0 0 4
24-Nov-22	1					Coding uC	5	Creating state machine with Unconnected, Connecting, and Connected states	Created new iSeat code with Connected state (same operation as before), Unconnected state (asks user to input network info to connected to), and Connecting state (connect to Network)	0 5 0 0 0 0
25-Nov-22			1			Edit Seats	2	Admin can edit Name and Location	Can be a feature that helps locate some and organise some of the seats	0 0 0 2 0 0
26-Nov-22				1		LOG feature for an Admin user	5	Admin user, can see a page with history of add/remove rooms, or renames, or when sensors get added/removed from rooms	1 string per log entry that displays "ROOM1 sensor deleted", "ROOM2 rename to ROOM3", "ROOM1 deleted", which shows only to the admin	0 0 0 0 5 0
27-Nov-22		1	1			User/Admin Login Verification	4	Admin and users now have a profile page they can see their email and displayed username	Admins and users can go to their profile and modify avatar for a unique experience	0 0 4 4 0 0
27-Nov-22	1					Preparing App for Demo, bug fixes	2	Integrating all new activities into a working whole on the app	App for demo	0 2 0 0 0 0
27-Nov-22	1	1	1	1	1	Prepare Final Oral Presentation	1			1 1 1 1 1 1
28-Nov-22	1	1	1	1	1	Prepare Final App Demo	1			1 1 1 1 1 1
7-Dec-22	1	1	1	1	1	Prepare Final Report Part 1	2			2 2 2 2 2 2
8-Dec-22	1	1	1	1	1	Prepare Final Report Part 2	2			2 2 2 2 2 2
Total hours total team hours								7 19 20 20 22 11		99
Total hour over project Total team hours over project								44.0 80.0 59.5 77.5 71.5 54.5		387

XIV. EXPECTATION OF ORIGINALITY FORM

Form ENCS-SAS (03/04)

Faculty of Engineering and Computer Science Expectations of Originality

This form has been created to ensure that all students in the Faculty of Engineering and Computer Science comply with principles of academic integrity prior to submitting coursework to their instructors for evaluation: namely reports, assignments, lab reports and/or software. All students should become familiar with the University's Code of Conduct (Academic) located at http://web2.concordia.ca/Legal_Counsel/policies/english/AC/Code.html

Please read the back of this document carefully before completing the section below. This form must be attached to the front of all coursework submitted to instructors in the Faculty of Engineering and Computer Science.

Course Number: COEN 390 **Instructor:** Dr. William E. Lynch

Type of Submission (Please check off responses to both a & b)

- | | | | |
|---|--|-------------------------------------|-----------------------------------|
| a. <input type="checkbox"/> Report | <input checked="" type="checkbox"/> Assignment | <input type="checkbox"/> Lab Report | <input type="checkbox"/> Software |
| b. <input type="checkbox"/> Individual submission | <input checked="" type="checkbox"/> Group Submission (All members of the team must sign below) | | |

Having read both sides of this form, I certify that I/we have conformed to the Faculty's expectations of originality and standards of academic integrity.

Name: Adnan Saab ID No: 40075504 Signature:  Date: 2022-12-10
(please print clearly)

Name: Samson Kaller ID No: 40136815 Signature:  Date: 2022-12-10
(please print clearly)

Name: Farah Salhany ID No: 40074803 Signature:  Date: 2022-12-10
(please print clearly)

Name: Shahin Khalkhali ID No: 40057384 Signature:  Date: 2022-12-10
(please print clearly)

Name: Shayan Khalkhali ID No: 40059491 Signature:  Date: 2022-12-10
(please print clearly)

Name: Marwan Al-Ghaziri ID No: 40126554 Signature:  Date: 2022-12-10
(please print clearly)

Do Not Write in this Space – Reserved for Instructor